### PROGRAMS on NUMBERS

#### Write a program to Print 1 to N numbers?

class Printnums

{

public static void main (String[] args)

{

java.util.Scanner sc = new java.util.Scanner (System.in); System.out.println ("enter value of n");

int n = sc.nextInt();

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

{

System.out.println (i);

}

}

}

###### OUTPUT:

enter value of n: 10 1

2

3

4

5

6

7

8

9

10

#### Write a program to Print REVERSE of N to 1 numbers?

class Printnums

{

public static void main(String[] args)

{

java.util.Scanner sc = new java.util.Scanner(System.in); System.out.println ("enter value of n");

int n=sc.nextInt(); for(int i=n ;i>=1;i--)

{

System.out.print(i);

}

}

}

###### OUTPUT:

enter value of n: 10 10 9 8 7 6 5 4 3 2 1

#### Write a program to display sum of 1 to N numbers?

class Sumnum

{

public static void main(String[] args)

{

java.util.Scanner sc=new java.util.Scanner(System.in); System.out.println("enter value of n");

int n=sc.nextInt(); int sum=0;

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

{

sum+=i;

}

System.out.println(sum);

}

}

###### OUTPUT:

enter value of n: 10 55

#### Write a program to check given number is EVEN or ODD?

class EvenOdd

{

public static void main(String[] args)

{

java.util.Scanner sc=new java.util.Scanner(System.in); System.out.println("enter the num");

int n=sc.nextInt(); if(n%2==0)

System.out.println(n+” is even");

else

}

}

System.out.println(n+" is odd");

###### OUTPUT:

enter the num: 20 20 is even

F:\Practice>java Even(Command prompt) enter the num: 11

11 is odd

#### Write a program to display PRIME NUMBERS from 1 to n?

class Prime

{

public static void main (String [] args)

{

java.util.Scanner sc=new java.util.Scanner (System.in); System.out.println ("enter number");

int n=sc.nextInt ();

System.out.println ("Prime numbers between 1 and " + n);

//loop through the numbers one by one for (int i=1; i < n; i++)

{

boolean isPrime = true;

//check to see if the number is prime for (int j=2; j < i ; j++)

{

if (i % j == 0)

{

isPrime = false; break;

}

}

// print the number if (isPrime)

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

}

}

}

#### OUTPUT:

enter number 25

Prime numbers between 1 and 25

1 2 3 5 7 11 13 17 19 23

#### Write a program to check whether the given number is PRIME or not?

class Prime

{

public static void main(String[] args)

{

java.util.Scanner sc=new java.util.Scanner(System.in); System.out.println("enter number");

int n=sc.nextInt(); int i;

if(n==1)

{

System.out.println("Prime starts from 2");

}

for(i=2;i<n ;i++)

{

if(n%i==0)

System.out.println("not a prime");

break;

}

if(n==i)

System.out.println("prime");

}

}

###### OUTPUT:

Enter the number : 17 Prime

#### Write a program to find SUM OF PRIME numbers?

import java.util.Scanner; public class SumofPrime

{

public static void main(String[] args)

{

Scanner scn=new Scanner(System.in);

System.out.println("Enter the range to print sum of prime Nos ");

int range=scn.nextInt(); int sum=0;

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

{

if(isPrime(i)) sum=sum+i;

}

System.out.println(sum);

}

public static boolean isPrime(int num)

{

if(num==1) return false; for(int i=2;i<num ;i++)

{

if(num%i==0)

{

return false;

}

}

return true;

}

}

###### OUTPUT:

Enter the range to print sum of prime Nos.....

10

17

#### Write a program to find SUM OF PRIME numbers?

**package** com.NumbersPrograms;

**import** java.util.Scanner;

**public** **class** PrimeSum {

// CHECK WHEATHER THE NUM IS PRIME OR NOT

**static** **boolean** isPrime(**int** num)

{

**if**(num==1)

{

**return** **false**;

}

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

{

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

{

**return** **false**;

}

}

**return** **true**;

}

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

{

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

System.***out***.println("Enter the range to sum of prime nos");

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

**int** sum=0;

**for**(**int** i=1;i<=range;i++)

{

**if**(*isPrime*(i))

{

sum=sum+i;

}

}

System.***out***.println("Sum of prime number between the given range "+sum);

}

}

#### Write a program to display MULTIPLICATION table?

class Multiplication

{

public static void main(String[] args)

{

java.util.Scanner sc=new java.util.Scanner(System.in); System.out.println("enter value of n");

int n=sc.nextInt(); for(int i=1;i<=10;i++)

{

System.out.println(n+"\*"+i+"="+(n\*i));

}

}

}

**Output:**

enter value of n: 2 2\*1=2

2\*2=4

2\*3=6

2\*4=8

2\*5=10

2\*6=12

2\*7=14

2\*8=16

2\*9=18

2\*10=20

#### Write a program to display MULTIPLICATION TABLES?

class Tables

{

public static void main(String[] args)

{

java.util.Scanner sc=new java.util.Scanner(System.in); System.out.println("enter value of n");

int n=sc.nextInt(); for(int i=1;i<=n ;i++)

{

for (int j=1;j<=10 ;j++ )

{

System.out.print(j+"\*"+i+"="+j\*i+"\t");

}

}

System.out.println();

}

}

#### OUTPUT:

enter value of n: 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1\*1=1 | 2\*1=2 | 3\*1=3 | 4\*1=4 | 5\*1=5 |
| 1\*2=2 | 2\*2=4 | 3\*2=6 | 4\*2=8 | 5\*2=10 |
| 1\*3=3 | 2\*3=6 | 3\*3=9 | 4\*3=12 | 5\*3=15 |
| 1\*4=4 | 2\*4=8 | 3\*4=12 | 4\*4=16 | 5\*4=20 |
| 1\*5=5 | 2\*5=10 | 3\*5=15 | 4\*5=20 | 5\*5=25 |
| 1\*6=6 | 2\*6=12 | 3\*6=18 | 4\*6=24 | 5\*6=30 |
| 1\*7=7 | 2\*7=14 | 3\*7=21 | 4\*7=28 | 5\*7=35 |
| 1\*8=8 | 2\*8=16 | 3\*8=24 | 4\*8=32 | 5\*8=40 |
| 1\*9=9 | 2\*9=18 | 3\*9=27 | 4\*9=36 | 5\*9=45 |
| 1\*10=10 | 2\*10=20 | 3\*10=30 | 4\*10=40 | 5\*10=50 |

#### Write program weather the number is PERFECT NUMBER or not?

**Def:**

**Perfect number,** a positive [integer](https://www.britannica.com/topic/integer) that is equal to the sum of its proper divisors. The smallest perfect number is 6, which is the sum of 1, 2, and 3.

import java.util.\*; class Perfectnumber

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter a number"); int num=sc.nextInt();

int sum=1;

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

{

if (num%i==0) sum=sum+i;

}

if (sum==num)

{

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

}

else

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

}

}

**OUTPUT:**

enter a number 6

6 is a Perfect number

#### Write a program to display RANGE of PERFECT NUMBERS?

import java.util.\*;

class Rangeperfectnumber

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter a number"); int n=sc.nextInt();

for(int num=1;num<=n; num++)

{

int sum=1;

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

{

if (num%i==0) sum=sum+i;

}

if (sum==num)

{

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

}

}

}

}

#### OUTPUT:

enter a number 100

1is a perfect number 6is a perfect number 28is a perfect number

#### Write a program to display sum of Strong Number in a given range?

**package** com.NumbersPrograms;

**import** java.util.Scanner;

**public** **class** StrongNumberSum

{

**static** **int** printFact(**int** n)

{

**int** fact=1;

**while**(n>0)

{

fact=fact\*n;

n--;

}

**return** fact;

}

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

{

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

System.***out***.println("Enter the first number");

**int** n1=sc.nextInt();

System.***out***.println("Enter the second number");

**int** n2=sc.nextInt();

System.***out***.println("Strong number between "+n1+" and "+n2+" is ");

**int** sum1=0;

**for**(**int** i=n1;i<=n2;i++)

{

**int** n3=i;

**int** sum=0;

**int** temp=n3;

**while**(n3!=0)

{

**int** rem=n3%10;

sum=sum+*printFact*(rem);

n3=n3/10;

}

**if**(sum==temp)

{

System.***out***.println(temp+" ");

sum1=sum1+temp;

}

}

System.***out***.println("Sum of Strong number "+sum1);

}

}

#### Write a program to check the given number is PALINDROME or not?

import java.util.\*; class Palindrome

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter a number"); int n =sc.nextInt();

int t=n; int rev=0;

while (n!=0)

{

rev=rev\*10+(n%10); n=n/10;

}

if (rev==t)

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

else

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

}

}

#### OUTPUT:

enter a number 121

121 is a palindrome number

enter a number 143

143 is not a palindrome number

#### Write a program to find palindrome number in a given range?

**package** com.NumbersPrograms;

**import** java.util.Scanner;

//WAP TO FIND PALINDROME NUMBERS IN A GIVEN RANGE

**public** **class** PalindromeRange

{

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

{

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

System.***out***.println("Enter first number");

**int** n1=sc.nextInt();

System.***out***.println("Enter the second number");

**int** n2=sc.nextInt();

System.***out***.println("Palindrome number between"+n1+"and"+n2+"are");

**for**(**int** i=n1;i<=n2;i++)

{

**int** n3=i;

**int** rev=0;

**while**(n3!=0)

{

**int** rem=n3%10;

rev=rev\*10+rem;

n3=n3/10;

}

**if**(rev==i)

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

}

}

}

#### Write a program to find sum of palindrome numbers in the given range?

**package** com.NumbersPrograms;

**import** java.util.Scanner;

**public** **class** PalindromeNumbersSum

{

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

{

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

System.***out***.println("Enter the first number");

**int** n1=sc.nextInt();

System.***out***.println("Enter the second number");

**int** n2=sc.nextInt();

System.***out***.println("Palindrome number between "+n1+" and "+n2+" is");

**int** sum=0;

**for**(**int** i=n1;i<=n2;i++)

{

**int** rev=0;

**int** n3=i;

**while**(n3!=0)

{

**int** rem=n3%10;

rev=rev\*10+rem;

n3=n3/10;

}

**if**(rev==i)

{

sum=sum+i;

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

}

}

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

System.***out***.println("sum is "+sum);

}

}

#### Write a program to find the FACTORIAL of a given number?

import java.util.\*; class Factorial

{

public static void main(String[] args)

{

Scanner scn=new Scanner(System.in); System.out.println("enter the number"); int n=scn.nextInt();

int fact=1;

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

{

fact=fact\*i;

}

System.out.println(fact);

}

}

###### OUTPUT:

Enter the number 5

120

#### Write a program to find the FACTORIAL of a given RANGE of numbers?

import java.util.\*; class FactRange

{

static int fact(int n)

{

int fact=1; while (n>0)

{

fact=fact\*n; n--;

}

return fact;

}

public static void main(String[] args)

{

Scanner scn=new Scanner(System.in); System.out.println("enter the factorial range number"); int k=scn.nextInt();

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

{

System.out.println(i+"! >"+fact(i));

}

}

}

###### OUTPUT:

enter the factorial range number :7 1! >1

2!---->2

3!---->6

4!---->24

5!---->120

6!---->720

7!---->5040

**Find the Number Of Trailing Zeros In A Factorial**

**package** com.NumberPrograms;

//Find the Number Of Trailing Zeros In A Factorial

//INPUT: N=5

//OUTPUT: 1

/\*LOGIC: N!-->N/5+N/25+N/125+N/625+N/3125+....

\* N!-->N/5^1+N/5^2+N/5^3+N/5^4+.....

\* LET, N=5

\* 5/5+5/25+5/125+...=1+0+0+...=1

\*/

**public** **class** FindFactorialTraillingZeros {

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

**int** n=5;

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

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

}

**private** **static** **int** traillingZeros(**int** num) {

**int** powerOf5=5;

**int** res=0;

**while**(num>=powerOf5)

{

res=res+(num/powerOf5);

powerOf5=powerOf5\*5;

}

**return** res;

}

**static** **int** findFact(**int** n)

{

**int** fact=1;

**while**(n!=0)

{

fact=fact\*n;

n--;

}

**return** fact;

}

}

**Find total Number Of Zeros In A Factorial**

**package** com.NumberPrograms;

//Find total Number Of Zeros In A Factorial

//INPUT:15

//OUTPUT:4

**public** **class** CountFactorialZeros {

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

**int** n=15;

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

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

}

**private** **static** **int** countZeros(**int** n) {

**int** num=*findFact*(n);

**int** count=0;

**while**(num!=0)

{

**int** rem=num%10;

**if**(rem==0)

count++;

num=num/10;

}

**return** count;

}

**private** **static** **int** findFact(**int** n) {

**int** fact=1;

**while**(n!=0)

{

fact=fact\*n;

n--;

}

**return** fact;

}

}

#### Write program to check the given number is STRONG or not?

#### OR

# Peterson Number in Java

**Def: Strong numbers** are the **numbers** whose sum of factorial of digits is equal to the original **number**. Example: 145 is a **strong number**.

import java.util.\*; class Strongnumber

{

static int fact(int n)

{

int fact=1; while (n>0)

{

fact= fact\*n; n--;

}

return fact;

}

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter a number"); int n =sc.nextInt();

int num=n; int sum=0; int t=num;

while (num!=0)

{

int r=num%10; sum=sum + fact(r); num=num/10;

}

if (sum==t)

System.out.println(t+" is a strong number");

else

}

}

System.out.println(t+" not a strong number"

**OUTPUT:**

enter a number 143

143not a strong number

#### Write program weather to find range of STRONG NUMBER?

import java.util.\*; class Strongnumber

{

static int fact(int n)

{

int fact=1; while (n>0)

{

fact= fact\*n; n--;

}

return fact;

}

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter a Range"); int n =sc.nextInt();

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

{

int num=i; int sum=0; int t=num;

while (num!=0)

{

int r=num%10; sum=sum + fact(r); num=num/10;

}

if (sum==t)

System.out.println(t+ " is a strong number");

}

}

}

###### OUTPUT:

enter a Range 145

1is a strong number 2is a strong number 145 is a strong number

#### Write a program to display FIBONACCI series of a number?

**Def:** a series of numbers in which each number ( *Fibonacci number* ) is the sum of the two preceding numbers. The simplest is the series 1, 1, 2, 3, 5, 8, etc.

class Fibonacci

{

static int fib(int n)

{

if(n==0)

return 0; if(n==1)

return 1;

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

}

public static void main(String[] args)

{

java.util.Scanner sc=new java.util.Scanner(System.in); System.out.println("Enter the number");

int m=sc.nextInt(); int f=fib(m); System.out.println(f);

}

}

###### OUTPUT:

Enter the number 10

55

#### Write a program to display range of FIBONACCI numbers?

import java.util.Scanner; public class FibonacciSeries1

{

public static void main(String[] args)

{

Scanner scn=new Scanner(System.in); System.out.println("enter the range: ");

int range=scn.nextInt(); int a=0;

int b=1; int c=0;

System.out.print(a); System.out.print(b);

for (int i = 2; i <=range; i++)

{

c=a + b;

}

}

}

###### OUTPUT:

if(c<=range)

{

//c=a + b; System.out.print(c); a=b;

b=c;

}

Enter the range….

50

0 1 1 2 3 5 8 13 21 34

#### Write a program to REVERSE the number?

import java.util.Scanner; class Reversenum

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter the number"); int num=sc.nextInt();

int t=num; int rev=0;

while(num!=0)

{

rev = rev\*10+(num%10); num = num/10;

}

System.out.println(rev);

}

}

###### OUTPUT:

enter the number 105

501

#### Write a program to display GCD of two numbers?

import java.util.Scanner; class Gcd

{

static int gcd(int m ,int n)

{

if(m<n)

return gcd(n ,m); if(n==0)

return m; return gcd(n, m%n);

}

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in); System.out.println(" Enter the two numbers"); int p = sc.nextInt();

int q = sc.nextInt();

int a=gcd(p, q); System.out.println(a);

}

}

###### OUTPUT:

Enter the two numbers 90

120

30

#### Write a program to check the given number is PRIME PALINDROME or not?

import java.util.\*; class Palindrome

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter a number"); int n =sc.nextInt();

int t=n; int rev=0; int i;

while (n!=0)

{

rev=rev\*10+(n%10); n=n/10;

}

if (rev==t)

{

for( i=2;i<rev ;i++)

{

if(rev % i==0)

{

break;

}

}

System.out.println("not a prime palindrome");

}

else

if(rev==i)

System.out.println(t+ "is a prime palindrome number");

System.out.println(t+ "is not a prime palindrome number");

}

}

###### OUTPUT:

enter a number 313

313 is a prime palindrome number

enter a number 103

103 is not a prime palindrome number

#### Write a Program to check the given number is ARMSTRONG or not?

##### **Def:** An Armstrong number is an integer such that the sum of the power of its digits is equal to the number itself.

For example, 371 is an Armstrong number since 3\*\*3 + 7\*\*3 + 1\*\*3 = 371.

##### 9 is an Armstrong number since 9\*1= 9.

import java.util.Scanner; public class Armstrong1

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter the number"); int n=sc.nextInt();

boolean r=isArmstrong(n);

if(r)

else

System.out.println("Given num is Armstrong"); System.out.println("Given num is not Armstrong");

}

static int countDigit(int num)

{

int count=0; while(num>0)

{

count++; num=num/10;

}

return count;

}

static int pow(int n, int p)

{

int pw=1; while(p>0)

{

pw=pw\*n; p--;

}

return pw;

}

static boolean isAmstrong(int x)

{

int nd=countDigit(x); int t=x;

int sum=0; while(t>0)

{

int r=t%10; sum=sum+ pow(r ,nd); t=t/10;

}

if(sum==x)

return true;

else

}

}

return false;

###### OUTPUT:

enter the number 153

Given num is Armstrong enter the number

1

Given num is Armstrong

#### Write a Program to display the range of ARMSTRONG numbers?

import java.util.Scanner; public class Armstrong2

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter the number"); int n=sc.nextInt();

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

{

boolean r=isAmstrong(i);

if(r)

System.out.println(i +" is Armstrong");

}

}

static int countDigit(int num)

{

int count=0; while(num>0)

{

count++; num=num/10;

}

return count;

}

static int pow(int n ,int p)

{

int pw=1; while(p>0)

{

pw=pw\*n; p--;

}

return pw;

}

static boolean isAmstrong(int x)

{

int nd=countDigit(x); int t=x;

int sum=0;

while(t>0)

{

int r=t%10;

sum=sum +pow(r ,nd); t=t/10;

}

if(sum==x)

return true;

else

}

}

return false;

###### OUTPUT:

enter the number: 300 0 is Armstrong

1. is Armstrong
2. is Armstrong
3. is Armstrong
4. is Armstrong
5. is Armstrong
6. is Armstrong
7. is Armstrong
8. is Armstrong
9. is Armstrong

153 is Armstrong

#### Write a program to Swap two numbers without using 3rd variable?

class Swap

{

public static void main(String[] args) { int i=10;

int j=20; i=i + j; j=i-j;

i=i-j;

System.out.println(“i=”+i); System.out.println(“j=”+j);

}

}

###### OUTPUT:

i=20 j=10

#### Write a program to Swap two numbers with using 3rd variable?

class Swapv

{

public static void main(String[] args)

{

int i=10; int j=20; int k; k=i;

i=j; j=k;

System.out.println(“i=”+i); System.out.println(“j=”+j);

}

}

###### OUTPUT:

i=20 j=10

#### Write a program to Swap two numbers ?

**package** com.NumbersPrograms;

**import** java.util.Scanner;

**public** **class** SwapTwoNumbers

{

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

{

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

System.***out***.println("Enter the value of a and b");

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

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

System.***out***.println("Before swaping "+a+" "+b);

SWAP TWO NUMBERS WITH USING THIRD VARIABLE

// int temp=a;

// a=b;

// b=temp;

SWAP TWO NUMBERS WITHOUT USING THIRD VARIABLE

// a=a+b;

// b=a-b;

// a=a-b;

SWAP TWO NUMBERS USING BITWISE OPERATORS

// a=a^b;

// b=a^b;

// a=a^b;

SWAP TWO NUMBERS USING MULTIPLICATION AND DIVISION

// a=a\*b;

// b=a/b;

// a=a/b;

System.***out***.println("after swaping "+a+" "+b);

}

}

#### Write a program to find biggest digit in a given number?

**package** com.Assignment1;

**import** java.util.Scanner;

// JAVA PROGRAM TO FIND LARGEST DIGIT FROM A GIVEN NUMBER

**public** **class** BiggestDigit

{

**static** **int** isBiggest(**int** n)

{

int largest=-9;//int largest=Integer.***MIN\_VALUE***;

**while**(n!=0)

{

**int** rem=n%10;

**if**(largest<rem)

{

largest=rem;

}

n=n/10;

}

System.***out***.println("largest digit is "+largest);

**return** largest;

}

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

{

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

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

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

*isBiggest*(n);

}

}

#### Write a program to find smallest digit in a given number?

**package** com.Assignment1;

**import** java.util.Scanner;

// JAVA PROGRAM SMALLEST DIGIT FROM A GIVEN NUMBER

**public** **class** SmallestDigit

{

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

{

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

System.***out***.println("Enter the number");

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

**int** min=9; //int min=Integer.MAX\_VALUE;

**while**(n!=0)

{

**int** rem=n%10;

**if**(rem<min)

{

min=rem;

}

n=n/10;

}

System.***out***.println("smallest number is "+min);

}

}

#### Write a program to find digit sum in a given number?

**package** com.Assignment1;

**import** java.util.Scanner;

// JAVA PROGRAM TO FIND THE DIGIT SUM IN A GIVEN NUMBER

**public** **class** DigitSum

{

**static** **int** printSum(**int** n)

{

**int** sum=0;

**while**(n>0)

{

**int** rem=n%10;

sum=sum+rem;

n=n/10;

}

System.***out***.println("Digit sum is "+sum);

**return** sum;

}

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

{

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

System.***out***.println("Enter the number");

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

*printSum*(n);

}

}

#### Write a program to find no of divisors in a given number?

**package** com.Assignment1;

**import** java.util.Scanner;

// JAVA PROGRAMS TO FIND NO OF DIVISORS/FACTORS OF A NUMBER

**public** **class** NoOfDivisor

{

**static** **void** printDiv(**int** n)

{

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

{

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

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

}

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

// if(n%i==0)

// System.out.println(i+" ");

// }

//PRINT THE NUMBER ITSELF TOO

// System.out.println(n);

}

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

{

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

System.***out***.println("Enter the number");

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

System.***out***.println("The divisor of "+n+" is");

*printDiv*(n);

}

}

#### Write a program to find total no of divisors in a given number?

**package** com.Assignment1;

**import** java.util.Scanner;

// PROGRAMS TO FIND TOTAL NUMBER OF DIVISORS IN A GIVEN NUMBER

**public** **class** CountDivisors

{

**static** **int** countDiv(**int** n)

{

// int count=0;

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

{

// if(n%i==0)

// count++;

// }

**int** count=1;

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

{

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

count++;

}

**return** count;

}

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

{

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

System.***out***.println("Enter the number");

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

**int** dv=*countDiv*(n);

System.***out***.println("Total number of divisors are "+dv);

}

}

**Program To Find All The Prime Factors Of A Number**

**package** com.NumberPrograms;

//Program To Find All The Prime Factors Of A Number

//INPUT:40 //INPUT:13

//OUTPUT:2 2 2 5 //OUTPUT:13

**public** **class** FindPrimeFactors {

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

**int** n=40;

*findPrimeFactors*(n);

}

**private** **static** **void** findPrimeFactors(**int** n) {

**int** i=2;

**while**(i\*i<=n)

{

**while**(n%i==0)

{

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

n=n/i;

}

i++;

}

**if**(n>1)//THIS CONDITION WILL EXECUTE ONLY FOR PRIME NUMBER

{

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

}

}

}

#### Write a program to check given number is prime or not?

**package** com.Assignment1;

**import** java.util.Scanner;

// CHECK PRIME NUMBER BY USING FACTORS/DIVISORS LOGIC

**public** **class** PrimeNum

{

**static** **boolean** isPrime(**int** n)

{

// int count=1;

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

{

// if(n%i==2)

// count++;

// }

// return count==2;

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

{

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

**return** **false**;

}

**return** **true**;

}

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

{

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

System.***out***.println("Enter the number");

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

**boolean** rs=*isPrime*(n);

**if**(rs==**true**)

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

**else**

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

}

}

#### Write a program to find difference between largest and smallest digit in a given number?

**package** com.Assignment2;

**import** java.util.Scanner;

// DIFFERENCE BETWEEN LARGEST AND SMALLEST DIGIT IN A GIVEN NUMBER

**public** **class** DifferenceDigit

{

**static** **int** diffDigit(**int** n)

{

**int** min=Integer.***MAX\_VALUE***;

**int** max=Integer.***MIN\_VALUE***;

**while**(n!=0)

{

**int** rem=n%10;

**if**(min>rem)

min=rem;

**if**(max<rem)

max=rem;

n=n/10;

}

**int** diff=max-min;

**return** diff;

}

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

{

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

System.***out***.println("Enter the number");

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

**int** d=*diffDigit*(n);

System.***out***.println("Difference is "+d);

}

}

#### Write a program to find average of digit in a given number?

**package** com.Assignment2;

**import** java.util.Scanner;

// PROGRAMS TO FIND AVERAGE OF DIGIT IN A GIVEN NUMBER

**public** **class** DigitAvg

{

**static** **double** avgDigit(**int** n)

{

**double** sum=0;

**double** count=0;

**while**(n!=0)

{

**int** rem=n%10;

sum=sum+rem;

n=n/10;

count++;

}

**double** avg=sum/count;

**return** avg;

}

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

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

System.***out***.println("Enter the number");

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

**double** average=*avgDigit*(n);

System.***out***.println("Digit average is "+average);

}

}

#### Write a program to find how many prime digit in a given number?

**package** com.Assignment2;

**import** java.util.Scanner;

// FIND HOW MANY PRIME DIGITS PRESENT IN A NUMBER

**public** **class** PrimeDigit

{

**static** **int** countPrimeDigit(**int** n)

{

**int** count=0;

**while**(n!=0)

{

**int** rem=n%10;

n=n/10;

**if**(rem==2||rem==3||rem==5||rem==7)

count++;

}

**return** count;

}

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

{

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

System.***out***.println("Enter the number");

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

**int** pd=*countPrimeDigit*(n);

System.***out***.println("Total number of prime digits is "+pd);

}

}

#### Write a program to find how many even digit in a given number?

**package** com.Assignment2;

**import** java.util.Scanner;

**public** **class** EvenDigit

{

**static** **int** countEvenDigit(**int** n)

{

**int** count=0;

**int** sum=0;

**while**(n!=0)

{

**int** rem=n%10;

n=n/10;

**if**(rem%2==0)

count++;

}

**return** count;

}

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

{

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

System.***out***.println("Enter the number");

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

**int** ed=*countEvenDigit*(n);

System.***out***.println("Total number of even digits are "+ed);

}

}

#### Write a program to define a method to return n^p(power of a number)?

**package** com.Assignment2;

**import** java.util.Scanner;

// POWER OF A NUMBER--BASE^EXPONENT OR POWER

**public** **class** PowerOfNumber

{

**static** **int** powerNumber(**int** base,**int** exponent)

{

**int** power=1;

**for**(**int** i=1;i<=exponent;i++)

{

power=power\*base;

}

**return** power;

}

**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** exponent=sc.nextInt();

**int** pow=*powerNumber*(base,exponent);

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

}

}

#### Write a program to find sum of even digit in a given number?

**package** com.Assignment2;

**import** java.util.Scanner;

// EVEN DIGIT SUM IN A GIVEN NUMBER

**public** **class** EvenDigitSum

{

**static** **int** isEvenDigitSum(**int** n)

{

**int** sum=0;

**while**(n!=0)

{

**int** rem=n%10;

**if**(rem%2==0)

sum=sum+rem;

n=n/10;

}

**return** sum;

}

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

{

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

System.***out***.println("Enter the number");

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

**int** s=*isEvenDigitSum*(n);

System.***out***.println("Sum of Even Digit is "+s);

}

}

#### Write a program to find diserium number in given range?

**package** com.Assignment3;

**import** java.util.Scanner;

// JAVA PROGRAMS TO FIND DISARIUM NUMBER IN GIVEN RANGE

/\*

A number is called Disarium number if the sum of its power of

the positions from left to right is equal to the number.

135--1^1+3^2+5^3=135

\*/

**public** **class** DisariumNumberRange

{

**static** **boolean** isDisarium(**int** n)

{

**int** sum=0;

**int** temp=n;

**int** dc=*countDigit*(n);

**while**(n!=0)

{

**int** rem=n%10;

sum=sum+*pow*(rem,dc);

dc--;

n=n/10;

}

**return** sum==temp;

}

**static** **int** pow(**int** n,**int** p)

{

**int** prod=1;

**while**(p>0)

{

prod=prod\*n;

p--;

}

**return** prod;

}

**static** **int** countDigit(**int** n)

{

**int** count=0;

**while**(n>0)

{

count++;

n=n/10;

}

**return** count;

}

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

{

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

System.***out***.println("Enter first number");

**int** n1=sc.nextInt();

System.***out***.println("Enter second number");

**int** n2=sc.nextInt();

System.***out***.println("Diserium number between "+n1+ " and "+n2+ " are");

**int** count=0;

**for**(**int** i=n1;i<=n2;i++)

{

**if**(*isDisarium*(i)==**true**)

{

count++;

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

}

}

System.***out***.println("Total number "+count);

}

}

#### Write a program to check given number is diserium number or not?

**package** com.Assignment3;

**import** java.util.Scanner;

**public** **class** DisariumNumber

{

**static** **boolean** isDisarium(**int** n)

{

**int** sum=0;

**int** temp=n;

**int** dc=*countDigit*(n);

**while**(n!=0)

{

**int** rem=n%10;

sum=sum+*pow*(rem,dc);

n=n/10;

dc--;

}

**return** sum==temp;

}

**static** **int** pow(**int** n,**int** p)

{

**int** prod=1;

**while**(p>0)

{

prod=prod\*n;

p--;

}

**return** prod;

}

**static** **int** countDigit(**int** n)

{

**int** count=0;

**while**(n>0) {

count++;

n=n/10;

}

**return** count;

}

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

{

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

System.***out***.println("Enter the number");

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

**boolean** rs=*isDisarium*(n);

**if**(rs==**true**)

System.***out***.println(n+" is Disarium number");

**else**

System.***out***.println(n+" is not Disarium number ");

}

}

#### Write a program to check given number is Happy number or not?

**package** com.Assignment4;

**import** java.util.Scanner;

// HAPPY NUMBER-- CALCULATE THE SUM OF THE SQUARES OF ITS DIGIT.

//IF IT RETURNS 1, THEN THE NUMBER IS CALLED HAPPY NUMBER

// 7=7^2=49 4^2+9^2=16+81=97 9^2+7^2=81+49=130 1^2+3^2+0^2=10 1^2+0^2=1

**public** **class** HappyNumber

{

**static** **boolean** isHappy(**int** n)

{

**while**(n>9)

{

**int** sum=0;

**do** {

**int** rem=n%10;

sum=sum+(rem\*rem);

n=n/10;

}**while**(n!=0);

n=sum;

}

**return** n==1||n==7;

}

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

{

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

System.***out***.println("Enter the number");

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

**boolean** rs=*isHappy*(n);

**if**(rs)

System.***out***.println(n+" is a Happy Number");

**else**

System.***out***.println(n+" is not a Happy Number");

}

}

#### Write a program to check given number is Neon number or not?

**package** com.Assignment4;

**import** java.util.Scanner;

//NEON NUMBER--A positive integer whose sum of digits of its square is equal to the number itself is called a neon number

// 9--9^=81--8+1=9

**public** **class** NeonNumber

{

**static** **boolean** isNeon(**int** n)

{

**int** x=n\*n;

**int** sum=0;

**while**(x!=0)

{

**int** rem=x%10;

sum=sum+rem;

x=x/10;

}

**if**(sum==n)

**return** **true**;

**else**

**return** **false**;

}

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

{

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

System.***out***.println("Enter the number");

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

**boolean** rs=*isNeon*(n);

**if**(rs==**true**)

System.***out***.println(n+" is a Neon Number");

**else**

System.***out***.println(n+" is not a Neon Number");

}

}

#### Write a program to check given number is Automorphic number or not?

**package** com.Assignment4;

**import** java.util.Scanner;

//AUTOMORPHIC NUMBER--- A number is called an automorphic number if and only if the square of the given number ends with the same number itself

// 25--25^2=625--ends with 25(number itself)

**public** **class** AutomorphicNumber

{

**static** **boolean** isAutomorphic(**int** n)

{

**int** s=n\*n;

**while**(n>0)

{

**int** ld=n%10;

**int** lsd=s%10;

//if((n%10)!=(s%10))

**if**(ld%10==s%10)

**return** **true**;

n=n/10;

s=s/10;

}

**return** **false**;

}

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

{

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

System.***out***.println("Enter the number");

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

**boolean** rs=*isAutomorphic*(n);

**if**(rs==**true**)

System.***out***.println(n+" is an Automorphic Number");

**else**

System.***out***.println(n+" is not an Automorphic Number");

}

}

#### Write a program to check given number is Buzz number or not?

**package** com.Assignment4;

**import** java.util.Scanner;

// BUZZ NUMBER--A number is called a buzz number if it is divisible by 7 or it ends with 7

// 7--7%10=7 and 7%7=0 (7 is a buzz number)

**public** **class** BuzzNumber

{

**static** **boolean** isBuzz(**int** n)

{

**if**(n%10==7||n%7==0)

**return** **true**;

**else**

**return** **false**;

}

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

{

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

System.***out***.println("Enter the number");

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

**boolean** rs=*isBuzz*(n);

**if**(rs==**true**)

System.***out***.println(n+" is a Buzz number");

**else**

System.***out***.println(n+" is not a Buzz number");

}

}

#### Write a program to check given number is Duck number or not?

**package** com.Assignment4;

**import** java.util.Scanner;

// DUCK NUMBER---A number that has at least one 0 ( but not at the beginning of the number ) is called a duck number

// 13025 is a duck number

// 01235 is not a duck number

**public** **class** DuckNumber

{

**static** **boolean** isDuck(**int** n)

{

**while**(n!=0)

{

**if**(n%10==0)

**return** **true**;

n=n/10;

}

**return** **false**;

}

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

{

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

System.***out***.println("Enter the number");

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

**boolean** rs=*isDuck*(n);

**if**(rs==**true**)

System.***out***.println(n+" is a Duck number");

**else**

System.***out***.println(n+" is not a Duck number");

}

}

#### Write a program to check given number is Xylem or Phloem number ?

**package** com.Assignment4;

**import** java.util.Scanner;

// XylemPhloem Number---A number N will be a xylem number if the sum of its extreme (first and last) digits is equal to the sum of mean (all digits except first and last) digits.

//If the sum of extreme digits is not equal to the sum of mean digits, the number is called phloem number.

// Xylem Number--(SUM OF EXTREAMS DIGITS=SUM OF MEANS DIGITS)--1201--(1+1)=(2+0)

// Phloem Number--(SUM OF EXTREAMS DIGITS !=SUM OF MEANS DIGITS)--1351--(1+1)!=(3+5)

**public** **class** XylemPhloemNumber

{

**static** **boolean** isXP(**int** n)

{

**int** es=0;

**int** ms=0;

**int** num=n;

**while**(num!=0)

{

**if**(num==n||num<10)

es=es+num%10;

**else**

ms=ms+num%10;

num=num/10;

}

**if**(es==ms)

**return** **true**;

**else**

**return** **false**;

}

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

{

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

System.***out***.println("Enter the number");

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

**boolean** rs=*isXP*(n);

**if**(rs==**true**)

System.***out***.println(n+" is a Xylem number");

**else**

System.***out***.println(n+" is a Phloem number");

}

}

**FIND THE FIRST AND LAST DIGIT OF A GIVEN NUMBER**

**package** com.NumberPrograms;

//FIND THE FIRST AND LAST DIGIT OF A GIVEN NUMBER

**public** **class** FindFirstAndLastDigit {

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

**int** n=564899;

*findFirstLastDigit*(n);

}

**private** **static** **void** findFirstLastDigit(**int** n) {

**int** fd=0;

**int** ld=0;

ld=n%10;

System.***out***.println("Last digit of number "+ld);

**while**(n!=0)

{

fd=n;

n=n/10;

}

System.***out***.println("First number of number "+fd);

}

}

**Middle Digit Of a Number in Java**

**package** com.NumberPrograms;

**import** java.util.\*;

//Middle Digit Number in Java

/\*

The middle digit of a number is that which is exactly in-mid of the given number N is called middle digit number.

If a number has two middle digits then print both the digits as the middle number.

Consider a number 86596. In the given number, 5 is exactly in-mid because

there are two digits are in the left and the same number of digits are in the right of 5.

Let's take another number 178903. In the given number, if we make pair of three-three digits,

there will no digits remain in-mid. If we make pair of two-two digits, we will get two digits in-mid.

\*/

**public** **class** FindMiddleDigit {

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

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

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

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

sc.close();

**int** absNum = Math.*abs*(num);

**int** count = 0;

// for (int i = absNum; i != 0; i /= 10) {

// count++;

// }

**while**(num!=0)

{

num=num/10;

count++;

}

**if** (count % 2 == 1)

{

**for** (**int** i = 1; i <= count / 2; i++)

{

absNum = absNum/10;

}

**int** mid = absNum % 10;

System.***out***.println("Middle number is: " + mid);

}

**else**

{

**for** (**int** i = 1; i < count / 2; i++)

{

absNum =absNum/10;

}

**int** mid = absNum % 100;

System.***out***.println("Middle digits are: " + mid);

}

}

}

**Java Program to Display Alternate Prime Numbers**

**package** com.NumberPrograms;

//Java Program to Display Alternate Prime Numbers

//Prime Numbers from 1 to 20-->2,3,5,7,11,13,17,19

//Alternative Prime Numbers From 1 to 20-->2 5 11 17 OR 3 7 13 19

**public** **class** PrintAlternativePrimeNumbers {

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

**int** n=20;

System.***out***.println("Alternative Prime numbers from 1 to "+n+" are");

*printAlternativePrime*(n);

}

**private** **static** **void** printAlternativePrime(**int** n) {

**int** temp=2;

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

{

**if**(*checkPrime*(i))

{

**if**(temp%2==0)

{

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

}

temp++;

}

}

}

**private** **static** **boolean** checkPrime(**int** n) {

**if**(n<=1)

**return** **false**;

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

{

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

{

**return** **false**;

}

}

**return** **true**;

}

}

**Find Nth Prime Number Java**

**package** com.NumberPrograms;

//nth Prime Number Java

**public** **class** FindNthPrimeNumber {

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

**int** n=3;

**int** p=*findNthPrime*(n);

System.***out***.println(n+"th prime number is "+p);

}

**private** **static** **int** findNthPrime(**int** n) {

**int** flag=0,count=0;

**for**(**int** i=2;i>0;i++)

{

flag=0;

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

{

**if**(i%j==0)

{

flag=1;

**break**;

}

}

**if**(flag==0)

{

count++;

}

**if**(count==n)

{

**return** i;

}

}

**return** 0;

}

}

**Finding Prime Numbers range - Using Sieve Of Eratosthenes Approach In Java**

**package** com.NumberPrograms;

//Finding Prime Numbers range - Using Sieve Of Eratosthenes Approach In Java

//The approach consumes the O(n) of the memory and its time complexity is O(rootnlog(logn)).

**public** **class** FindingPrimeRange {

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

**int** n=10;

System.***out***.println("Prime number range from 1 to "+n);

*primeRange*(n);

}

**private** **static** **void** primeRange(**int** n) {

**boolean** prime[]=**new** **boolean** [n+1];

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

{

**if**(prime[i]==**false**)

{

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

{

prime[j]=**true**;

}

}

}

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

{

**if**(prime[i]==**false**)

{

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

}

}

}

}

**Java program to find square root of a Number without sqrt method**

**package** com.NumberPrograms;

//Java program to find square root of a Number without sqrt method

**public** **class** FindSqrtNumber {

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

**int** n=16;

**double** num=*findSqrt*(n);

System.***out***.println("Square root of "+n+" is "+num);

}

**private** **static** **double** findSqrt(**int** n) {

**double** temp=0;

**double** sr=n/2;

**do** {

temp=sr;

sr=(temp+(n/temp))/2;

}**while**((temp-sr)!=0);

**return** sr;

}

}

**Harshad number or not using Java**

**package** com.NumberPrograms;

//Harshad number or not using Java

//A Harshad Number is a number that is divisible by the sum of it's digits

**public** **class** HarshadNumber {

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

**int** n=213;

*checkHarshadNumber*(n);

}

**private** **static** **void** checkHarshadNumber(**int** n) {

**int** result=0;

**int** temp=n;

**while**(n!=0)

{

**int** rem=n%10;

result=result+rem;

n=n/10;

}

**if**(temp%result==0)

{

System.***out***.println(temp+" is a Harshad Number");

}

**else**

{

System.***out***.println(temp+" is not a Harshad Number");

}

}

}

**Check Whether or Not the Number is an Abundant Number in Java**

**package** com.NumberPrograms;

//Check Whether or Not the Number is an Abundant Number in Java

//A Number that is smaller than the sum of all it's factors except the number itself is known as an Abundant number.

/\*Example:

Input : Number = 18

Output : Yes, It's an Abundant Number

Explanation : The Factors for the number 18 are, 1, 2, 3, 6 and 9. We don't want to include the number itself.

Now the sum of the factors except the number itself is :

1 + 2 + 3 + 6 + 9 = 21

as the number 21>18 , the number itself.

It's an abundant number.

\*/

**public** **class** AbundantNumber {

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

**int** n=25;

*checkAbundantNumber*(n);

}

**private** **static** **void** checkAbundantNumber(**int** n) {

**int** sum=0;

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

{

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

{

sum=sum+i;

}

}

**if**(sum>n)

{

System.***out***.println(n+" is an Abundant Number");

System.***out***.println("The Abundance is: " + (sum - n));

}

**else**

{

System.***out***.println(n+" is not an Abundant Number");

}

}

}

**Friendly pair or not (amicable or not) using Java**

**package** com.NumberPrograms;

//Friendly pair or not (amicable or not) using Java

/\*The numbers whose ( sum of divisors ) / number ratio is same are known as Friendly Pair Numbers.

Example

Input : 6 28

Output : Yes, they are a friendly pair

Explanation : The factors of 6 and 28 except the numbers themselves are 1, 2, 3 and 1, 2, 4, 7, 14 respectively.

Now the sum of factors of both the numbers are 6 and 28 respectively.

When we divide the sums with the numbers we get 1 and 1 respectively.

As the ratio of both the number match, they are considered as a friendly pair.

\*/

**public** **class** FriendlyPairNumber {

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

**int** num1=6;

**int** num2=18;

**int** sum1=*getDivisors*(num1);

**int** sum2=*getDivisors*(num2);

**if**(sum1/num1==sum2/num2)

System.***out***.println(num1+" and "+num2+" both are friendly pair number");

**else**

System.***out***.println(num1+" and "+num2+" both are not friendly pair number");

}

**private** **static** **int** getDivisors(**int** n) {

**int** sum=0;

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

{

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

{

sum=sum+i;

}

}

**return** sum;

}

}

**CHECK THE GIVEN NUMBER IS PERFECT SQUARE OR NOT**

**package** com.NumberPrograms;

//CHECK THE GIVEN NUMBER IS PERFECT SQUARE OR NOT

**public** **class** CheckPerfectSquareNumber {

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

**int** n=169;

**boolean** rs=*isPerfectSquare*(n);

**if**(rs)

System.***out***.println(n+" is a perfect square number");

**else**

System.***out***.println(n+" is not a perfect square number");

}

**private** **static** **boolean** isPerfectSquare(**int** n) {

**for**(**int** i=0;i<=n/2;i++)

{

**if**(i\*i==n)

**return** **true**;

}

**return** **false**;

}

}

**Program To Find Square Root/Floor Of Square Root Of A Number (USING BINARY SEARCH)**

**package** com.NumberPrograms;

//Program To Find Square Root/Floor Of Square Root Of A Number(USING BINARY SEARCH)

//TIME COMPLEXCITY-O(logn)

**public** **class** SquareRootOfNumber {

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

**int** n=25;

**int** sqrt=*findSqrt*(n);

System.***out***.println("Square Root of "+n+" is "+sqrt);

}

**private** **static** **int** findSqrt(**int** n) {

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

{

**return** n;

}

**int** l=2;

**int** h=n/2;

**int** result=0;

**while**(l<=h)

{

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

**if**(mid\*mid==n)

{

**return** mid;

}

**else** **if**(mid\*mid<n)

{

l=mid+1;

result=mid;//FOR FINDING FLOOR OF SQUARE ROOT OF A NUMBER

}

**else**

{

h=mid-1;

//result=mid;//FOR FINDING CEIL OF SQUARE ROOT OF A NUMBER

}

}

**return** result;

}

}

**Sunny Number in Java**

**package** com.NumberPrograms;

//Sunny Number in Java

/\* A number is called a sunny number if the number next to the given number is a perfect square.

In other words, a number N will be a sunny number if N+1 is a perfect square.

Given, N=80 then N+1 will be 80+1=81, which is a perfect square of the number 9.

Hence 80 is a sunny number.

\*/

**public** **class** SunnyNumber {

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

**int** n=80;

*isSunnyNumber*(n);

}

**private** **static** **void** isSunnyNumber(**int** n) {

**if**(*isPerfectSquare*(n+1))

System.***out***.println(n+" is a Sunny Number");

**else**

System.***out***.println(n+" is not a Sunny Number");

}

**private** **static** **boolean** isPerfectSquare(**int** n) {

**for**(**int** i=0;i<=n/2;i++)

{

**if**(i\*i==n)

**return** **true**;

}

**return** **false**;

}

}

**Tech Number in Java**

**package** com.NumberPrograms;

//Tech Number in Java

/\* A number is called a tech number if the given number has an even number of digits

and the number can be divided exactly into two parts from the middle.

After equally dividing the number, sum up the numbers and find the square of the sum.

If we get the number itself as square, the given number is a tech number, else, not a tech number.

For example, 2025 is a tech number.

2025-->20|25-->20+25=45-->(45)^2=2025

\*/

**public** **class** TechNumber {

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

**int** n=2025;

**boolean** rs=*isTechNumber*(n);

**if**(rs)

System.***out***.println(n+"is a Tech Number");

**else**

System.***out***.println(n+"is not a Tech Number");

}

**private** **static** **boolean** isTechNumber(**int** n) {

**int** first=0,last=0,sum=0;

first=n%100;

last=n/100;

sum=first+last;

**if**(sum\*sum==n)

{

**return** **true**;

}

**return** **false**;

}

}

**FIND ALL TECH NUMBERS IN GIVEN RANGE**

**package** com.NumberPrograms;

//FIND ALL TECH NUMBERS IN GIVEN RANGE

**import** java.util.Scanner;

**public** **class** TechNumberRange {

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

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

System.***out***.println("Enter first number");

**int** n1=sc.nextInt();

System.***out***.println("Enter second number");

**int** n2=sc.nextInt();

System.***out***.println("Tech Number between "+n1+" and "+n2+" are");

**for**(**int** i=n1;i<=n2;i++)

{

**if**(*isSunny*(i))

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

}

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

}

**private** **static** **boolean** isSunny(**int** n) {

**int** first=0,last=0,sum=0;

first=n%100;

last=n/100;

sum=first+last;

**if**(sum\*sum==n)

{

**return** **true**;

}

**return** **false**;

}

}

# NUMBER CONVERSIONS

#### Write a program to convert BINARY to DECIMAL?

import java.util.\*; public class Bintodec

{

public static void main(String[] args)

{

System.out.println("enter the binary number"); Scanner sc=new Scanner(System.in);

long n =sc. nextLong(); long dec=0;

int count=0; while(n>0)

{

long r=n%10;

dec=dec +r\*pow(2,count); count++;

n/=10;

}

System.out.println("decimal Equivalent:" +dec);

}

static int pow(int n, int p)

{

int pw=1; while(p>0)

{

pw=pw\*n; p--;

}

return pw;

}

}

###### OUTPUT:

enter the binary number 111100001111

decimal Equivalent:3855

#### Write a program to convert DECIMAL to BINARY?

import java.util.\*; public class Dectobin

{

public static void main(String[] args)

{

System.out.println("enter the decimal number"); Scanner sc=new Scanner(System.in);

int n=sc.nextInt(); String bin=""; while(n>0)

{

int r=n%2; bin= r + bin; n=n/2;

}

System.out.println("Binary Equivalent:" + bin);

}

}

###### OUTPUT:

enter the decimal number 3855

Binary Equivalent:111100001111

#### Write a program to convert OCTAL to DECIMAL?

import java.util.\*; public class Octtodec

{

public static void main(String[] args)

{

System.out.println("enter the octal number"); Scanner sc=new Scanner(System.in);

int n =sc.nextInt(); int dec=0;

int count=0; while(n>0)

{

int r=n%10;

dec=dec + r\*pow(8,count); count++;

n/=10;

}

System.out.println("decimal Equivalent:" +dec);

}

static int pow(int n, int p)

{

int pw=1; while(p>0)

{

pw=pw\*n; p--;

}

return pw;

}

}

###### OUTPUT:

enter the octal number 763

decimal Equivalent:499

#### Write a program to convert DECIMAL to OCTAL?

import java.util.\*; public class Dectooct

{

public static void main(String[] args)

{

System.out.println("enter the decimal number"); Scanner sc=new Scanner(System.in);

int n=sc.nextInt(); String oct=""; while(n>0)

{

int r=n%8; oct= r + oct; n=n/8;

}

System.out.println("Octal Equivalent:" + oct);

}

}

###### OUTPUT:

enter the decimal number 56

Octal Equivalent:70

#### Write a program to convert DECIMAL to HEXADECIMAL?

import java.util.\*; public class Dectohex

{

public static void main(String[] args)

{

System.out.println("enter the decimal number"); Scanner sc=new Scanner(System.in);

int n=sc.nextInt(); String hex=""; while(n>0)

{

int r=n%16; switch (r)

{

case 10: hex='A'+ hex; break;

case 11: hex='B'+ hex;

break; case 12: hex='C'+ hex;

break; case 13: hex='D'+ hex;

break; case 14: hex='E'+ hex;

break; case 15: hex='F'+ hex;

break;

default: hex=r + hex; break;

}

n=n/16;

}

System.out.println("Hexadecimal Equivalent :"+hex);

}

}

###### OUTPUT:

enter the decimal number 469

Hexadecimal Equivalent :1D5

#### Write a program to convert DECIMAL to ALL(Octal , Hexa and Binary)?

import java.util.\*; public class DectoAll

{

public static void main(String[] args)

{

System.out.println("enter the number"); Scanner sc=new Scanner(System.in);

int n=sc.nextInt(); System.out.println("enter the base"); int ba=sc.nextInt();

System.out.println(ba +"base equivalent "+Convert(n, ba));

}

static String Convert(int num, int base)

{

String st="0123456789ABCDEF";

String b=""; while(num>0)

{

int r= num % base; b=st.charAt(r)+b; num=num/base;

}

return b;

}

}

###### OUTPUT:

enter the number: 469 enter the base: 16

16 base equivalent: 1D5

enter the number: 369 enter the base: 8

8 base equivalent : 561

enter the number: 50 enter the base: 2

2 base equivalent: 110010

#### Write a program to convert DECIMAL to HEXADECIMAL?

import java.util.Scanner; class HexatoDec

{

public static void main(String[] args)

{

System.out.println("enter the Hexa dec number"); Scanner sc=new Scanner(System.in);

String st=sc.nextLine(); int dec = 0;

int count = 0;

int l = st.length(); while(l>0)

{

int r=0;

char ch=st.charAt(l-1); if(ch>=65 && ch<=70) r=ch-55;

else if(ch>=97 && ch<=102) r=ch-87;

else

r=ch-48;

dec=dec + r\*pow(16,count); count++;

l--;

}

System.out.println("Decimal Equivalent: "+dec);

}

static int pow(int n ,int p)

{

int pw=1; while(p>0)

{

pw=pw\*n; p--;

}

return pw;

}

}

###### OUTPUT:

enter the Hexa dec number: 1D5 Decimal Equivalent: 469

**Binary to Decimal conversion**

**package** com.NumberConversion;

**import** java.util.\*;

//Binary to Decimal conversion

**public** **class** BinToDec {

**static** **int** pow(**int** n, **int** p) {

**int** pw=1;

**while**(p>0) {

pw=pw\*n;

p--;

}

**return** pw;

}

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

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

System.***out***.println("Enter the binary number");

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

**int** dec=0;

**int** count=0;

**while**(n>0) {

**int** r=n%10;

dec= dec+r\**pow*(2,count);

count++;

n=n/10;

}

System.***out***.println("Decimal is "+dec);

}

}

/\*

int binToDec(int bin) {

int dec=0;

int p=1;

while(bin!=0) {

int d=bin%10;

dec=dec+d\*p;

p=p\*2;

bin=bin/10;

}

return dec;

}

\*/

**Binary to HexaDecimal conversion**

**package** com.NumberConversion;

**import** java.util.Scanner;

//Binary to HexaDecimal conversion

**public** **class** BinToHexadecimal {

**static** **int** binToDec(**int** n) {

**int** dec=0;

**int** c=0;

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

**int** rem=n%10;

dec=dec+rem\**pow*(2,c);

c++;

n=n/10;

}

**return** dec;

}

**static** **int** pow(**int** n,**int** p) {

**int** pw=1;

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

pw=pw\*n;

p--;

}

**return** pw;

}

**static** String decToHexadecimal(**int** n) {

**int** dec=*binToDec*(n);

String hex="";

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

**int** rem=dec%16;

**if**(rem<10)

hex=rem+hex;

**else**

hex=(**char**)(rem+55)+hex;

dec=dec/16;

}

**return** hex;

}

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

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

System.***out***.println("Enter the binary number");

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

String num= *decToHexadecimal*(n);

System.***out***.println("Equivalent hexadecimal number is "+num);

}

}

**Binary to Octal conversion**

**package** com.NumberConversion;

**import** java.util.Scanner;

//Binary to Octal conversion

**public** **class** BinToOctal {

**static** **int** binToDec(**int** n) {

**int** dec=0;

**int** c=0;

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

**int** rem=n%10;

dec=dec+rem\**pow*(2,c);

c++;

n=n/10;

}

**return** dec;

}

**static** String decToOctal(**int** n) {

**int** dec=*binToDec*(n);

String str="";

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

**int** rem=dec%8;

str=rem+str;

dec=dec/8;

}

**return** str;

}

**static** **int** pow(**int** n,**int** p ) {

**int** pw=1;

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

pw=pw\*n;

p--;

}

**return** pw;

}

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

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

System.***out***.println("Enter the binary number");

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

String num=*decToOctal*(n);

System.***out***.println("Equivalent octal number is "+num);

}

}

**Decimal to Binary conversion**

**package** com.NumberConversion;

**import** java.util.Scanner;

// Decimal to Binary conversion

**public** **class** DecimalToBinary {

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

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

System.***out***.println("Enter decimal number");

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

String bin="";

**while**(n>0) {

**int** r=n%2;

bin=r+bin;

n=n/2;

}

System.***out***.println("Binary number is "+bin);

}

}

**Decimal to HexaDecimal conversion**

**package** com.NumberConversion;

**import** java.util.Scanner;

// Decimal to HexaDecimal conversion

**public** **class** DecimalToHexadecimal {

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

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

System.***out***.println("Enter decimal number");

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

String hex="";

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

**int** r=n%16;

**switch**(r) {

**case** 10:hex="A"+hex;

**break**;

**case** 11:hex="B"+hex;

**break**;

**case** 12:hex="C"+hex;

**break**;

**case** 13:hex="D"+hex;

**break**;

**case** 14:hex="E"+hex;

**break**;

**case** 15:hex="F"+hex;

**break**;

**default**:hex=r+hex;

**break**;

}

n=n/16;

}

System.***out***.println("Hexadecimal value is "+hex);

}

}

/\*

String decToHex(int n){

String hex="";

while(n!=0) {

int r=n%16;

if(r<10)

hex=r+hx;

else

hex=(char)(r+55)+hex;

n=n/16;

}

return hex;

}

\*/

**Decimal to Octal conversion**

**package** com.NumberConversion;

**import** java.util.Scanner;

//Decimal to Octal conversion

**public** **class** DecimalToOctal {

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

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

System.***out***.println("Enter the decimal number");

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

String str="";

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

**int** r=n%8;

str=r+str;

n=n/8;

}

System.***out***.println("Octal number is "+str);

}

}

**Octal to Binary conversion**

**package** com.NumberConversion;

**import** java.util.Scanner;

//Octal to Binary conversion

**public** **class** OctalToBinary {

**static** **int** octToDec(**int** n) {

**int** count=0;

**int** dec=0;

**while**(n>0) {

**int** r=n%10;

dec=dec+r\**pow*(8,count);

count++;

n=n/10;

}

**return** dec;

}

**static** String decToBinary(**int** n) {

**int** d=*octToDec*(n);

String bin="";

**while**(d>0) {

**int** r=d%2;

bin=r+bin;

d=d/2;

}

**return** bin;

}

**static** **int** pow(**int** n,**int** p) {

**int** pw=1;

**while**(p>0) {

pw=pw\*n;

p--;

}

**return** pw;

}

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

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

System.***out***.println("Enter the octal number");

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

String num=*decToBinary*(n);

System.***out***.println("Equivalent binary number is:"+num);

}

}

**Octal to Decimal conversion**

**package** com.NumberConversion;

**import** java.util.Scanner;

//Octal to Decimal conversion

**public** **class** OctalToDecimal {

**static** **int** pow(**int** n,**int** p) {

**int** pw=1;

**while**(p>0) {

pw=pw\*n;

p--;

}

**return** pw;

}

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

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

System.***out***.println("Enter the octal number");

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

**int** count=0;

**int** dec=0;

**while**(n>0) {

**int** r=n%10;

dec=dec+r\**pow*(8,count);

count++;

n=n/10;

}

System.***out***.println("Decimal is"+dec);

}

}

/\*

int octToDec(int oct) {

int dec=0;

int pw=1;

while(oct!=0) {

int d=oct%10;

dec=dec+d\*pw;

pw=pw\*8;

oct=oct/10;

}

return dec;

}

\*/

**Octal to HexaDecimal conversion**

**package** com.NumberConversion;

**import** java.util.Scanner;

//Octal to HexaDecimal conversion

**public** **class** OctalToHexadecimal {

**static** **int** octToDec(**int** n) {

**int** count=0;

**int** dec=0;

**while**(n>0) {

**int** r=n%10;

dec=dec+r\**pow*(8,count);

count++;

n=n/10;

}

**return** dec;

}

**static** String decToHexadecimal(**int** n) {

**int** dec=*octToDec*(n);

String hex="";

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

**int** rem=dec%16;

**if**(rem<10)

hex=rem+hex;

**else**

hex=(**char**)(rem+55)+hex;

dec=dec/16;

}

**return** hex;

}

**static** **int** pow(**int** n,**int** p) {

**int** pw=1;

**while**(p>0) {

pw=pw\*n;

p--;

}

**return** pw;

}

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

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

System.***out***.println("Enter the octal number");

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

String num=*decToHexadecimal*(n);

System.***out***.println("Equivalent hexadecimal number is:"+num);

}

}

***PROGRAMS on STAR PATTERNS***

#### Write a program to display EQUILATERAL TRIANGLE with stars?

import java.util.Scanner; public class EquiTri

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in); System.out.println("enter the number");

int n = sc.nextInt(); for(int i=0;i<n ;i++)

{

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

{

System.out.print(" ");

}

for(int k=0;k<=i; k++)

{

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

}

System.out.println( );

}

}

}

###### OUTPUT:

**enter the number: 7**

###### \*

**\* \***

###### \* \* \*

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

###### \* \* \* \* \*

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

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

#### Write a program to Display INVERTED TRIANGLE with stars?

import java.util.Scanner; public class InverTri

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in); System.out.println("enter the number");

int n = sc.nextInt(); for(int i=0;i<n ;i++)

{

for (int j=0;j<i; j++)

{

System.out.print(" ");

}

for(int k=0;k<2\*(n-i)-1;k++)

{

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

}

System.out.println ( );

}

}

}

###### OUTPUT:

enter the number: 4

###### \*\*\*\*\*\*\*

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

###### \*\*\*

**\***

#### Write a program to display the FILLED BOX with stars?

class FilledBox

{

public static void main(String[] args)

{

java.util.Scanner sc=new java.util.Scanner(System.in); System.out.println("enter value of n");

int n=sc.nextInt(); for(int i=1;i<n ;i++)

{

for (int j=0;j<n ;j++ )

{

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

}

System.out.println();

}

}

}

###### Output:

enter value of n: 7

###### \*\*\*\*\*\*\*

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

###### \*\*\*\*\*\*\*

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

###### \*\*\*\*\*\*\*

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

#### Write a program to display the HALLOW BOX with stars?

class Box1

{

public static void main(String[] args)

{

java.util.Scanner sc = new java.util.Scanner(System.in); System.out.println ("enter value of n");

int n = sc.nextInt(); for (int i=0;i<n ;i++ )

{

for (int j=0;j<n ;j++ )

{

if (i==0||j==0||i==n-1||j==n-1)

{

}

else

{

}

}

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

System.out.print(" ");

System.out.println();

}

}

}

###### Output:

enter value of n 7

###### \* \*\*\*\*\* \*

**\* \***

###### \* \*

**\* \***

###### \* \*

**\* \***

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

#### Write a program to display the BOX and CROSS inside it with stars?

class Box1

{

public static void main(String[] args)

{

java.util.Scanner sc=new java.util.Scanner(System.in); System.out.println("enter value of n");

int n=sc.nextInt(); for (int i=0;i<n ;i++ )

{

for (int j=0;j<n ;j++ )

{

if (i==0||j==0||i==n-1||j==n-1||i==j||i+j==n-1)

{

}

else

{

}

}

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

System.out.print(" ");

System.out.println();

}

}

}

###### OUTPUT:

enter value of n: 7

###### \*\*\*\*\*\*\*

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

###### \* \* \* \*

**\* \* \***

###### \* \* \* \*

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

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

#### Write a program to display CROSS mark with stars?

class Cross

{

public static void main(String[] args)

{

java.util.Scanner sc=new java.util.Scanner(System.in); System.out.println("enter value of n");

int n=sc.nextInt(); for(int i=1;i<n ;i++)

{

for (int j=0;j<n ;j++ )

{

if(i==j||I + j==n-1)

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

else

}

System.out.print(" ");

System.out.println();

}

}

}

###### OUTPUT:

**enter value of n 7(odd)**

###### \* \*

**\* \***

###### \*

**\* \***

###### \* \*

**\* \***

#### Write a program to display RIGHT ANGLE triangle with stars*?*

class Triangle

{

public static void main(String[] args)

{

java.util.Scanner sc=new java.util.Scanner(System.in); System.out.println("enter value of n");

int n=sc.nextInt(); for(int i=1;i<n ;i++)

{

for (int j=0;j<i ;j++ )

{

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

}

System.out.println();

}

}

}

###### OUTPUT:

**enter value of n :7**

###### \*

**\*\***

###### \*\*\*

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

###### \*\*\*\*\*

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

#### Write a program to display Reverse Triangle with stars*?*

class Triangle1

{

public static void main (String [] args)

{

java.util.Scanner sc=new java.util.Scanner (System.in); System.out.println ("enter value of n");

int n=sc.nextInt (); for (int i=1; i<n; i++)

{

for (int j=0; j<n; j++)

{

if (i<=j) else

}

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

System.out.print (" ");

System.out.println ();

}

}

}

OUTPUT:

enter value of n 7

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

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

#### Write a program to display MIRROR of RIGHT ANGLE triangle with stars*?*

class Triangle2

{

public static void main(String[] args)

{

java.util.Scanner sc=new java.util.Scanner(System.in); System.out.println("enter value of n");

int n=sc.nextInt(); for(int i=1;i<n ;i++)

{

for (int j=0;j<n ;j++ )

{

if(i + j>n-1)

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

else

}

System.out.print(" ");

System.out.println();

}

}

}

***OUTPUT:***

enter value of n: 7

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

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

#### Write a program to display DOWNWARD MIRROR of RIGHT ANGLE triangle with stars*?*

class Triangle2

{

public static void main(String[] args)

{

java.util.Scanner sc=new java.util.Scanner(System.in); System.out.println("enter value of n");

int n=sc.nextInt(); for(int i=1;i<n ;i++)

{

for (int j=0;j<n ;j++ )

{

if(i + j<=n-1)

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

else

}

System.out.print(" ");

System.out.println();

}

}

}

###### OUTPUT:

**enter value of n: 7**

###### \*\*\*\*\*\*

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

###### \*\*\*\*

**\*\*\***

###### \*\*

**\***

#### Write a program to display DIAMOND with stars?

class Diamond

{

public static void main(String[] args)

{

java.util.Scanner scn=new java.util.Scanner (System.in); System.out.println ("enter odd number");

int n=scn.nextInt(); int spaces=n/2;

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

{

for( int j=1;j<=spaces ;j++)

{

System.out.print(" ");

}

for ( int k=1;k<=stars ;k++)

{

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

}

System.out.println(); if (i<=n/2)

{

}

else

spaces--; stars+=2;

{

spaces++;

stars-=2;

}

}

}

}

OUTPUT:

\*

\*\*\*

\*\*\*\*\*

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

\*\*\*\*\*

\*\*\*

\*

#### Write a program to display HALLOWDIAMOND with stars?

import java.util.Scanner; class HallowDiamond

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in); System.out.println("enter the value of n"); int n = sc.nextInt();

n = (n+1)/2;

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

{

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

{

System.out.print(" ");

}

for (int j=0;j<2\*i+1 ;j++ )

{

if (j==0||j==2\*i)

{

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

}

else

}

System.out.print(" ");

}

n = n-1;

System.out.println();

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

{

for (int j=0;j<=i ;j++ )

{

System.out.print(" ");

}

for (int j=0;j<2\*(n-i)-1 ;j++ )

{

if (j==0||j==2\*(n-i)-2)

{

}

else

}

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

System.out.print(" ");

System.out.println();

}

}

}

###### OUTPUT:

**enter the value of n ; 13**

###### \*

**\* \***

###### \* \*

**\* \***

###### \* \*

**\* \***

###### \* \*

**\* \***

###### \* \*

**\* \***

###### \* \*

**\* \***

**\***

#### Write a program to display NUMBERS in DIAMOND shape?

import java.util.Scanner; class NumDiamond

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

System.out.println("enter the value of n");

int n = sc.nextInt(); n = (n+1)/2;

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

{

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

{

System.out.print(" ");

}

int k=1;

for (int j=0;j<2\*i+1 ;j++ )

{

System.out.print(""+k); if (j<(2\*i+1)/2)

k++;

else k--;

}

System.out.println();

}

n = n-1;

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

{

for (int j=0;j<=i ;j++ )

{

System.out.print(" ");

}

int k=1;

for (int j=0;j<2\*(n-i)-1 ;j++ )

{

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

if (j<(2\*(n-i)-1)/2) k++;

else

k--;

}

System.out.println();

}

}

}

###### OUTPUT:

enter the value of n: 7 1

121

12321

1234321

12321

121

1

#### Write a program to display CHARACTERS in DIAMOND shape?

import java.util.Scanner; class CharDiamond

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in); System.out.println("enter the value of n"); int n = sc.nextInt();

n = (n+1)/2; char ch='A';

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

{

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

{

System.out.print(" ");

}

int k=0;

for (int j=0;j<2\*i+1 ;j++ )

{

System.out.print(""+(char)(ch + k)); if (j<(2\*i+1)/2)

k++;

else

k--;

}

System.out.println();

}

n = n-1;

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

{

for (int j=0;j<=i ;j++ )

{

System.out.print(" ");

}

int k=0;

for (int j=0;j<2\*(n-i)-1 ;j++ )

{

System.out.print(""+(char)(ch + k));

if (j<(2\*(n-i)-1)/2) k++;

else

k--;

}

System.out.println();

}

}

}

###### OUTPUT:

enter the value of n: 7 A

ABA ABCBA ABCDCBA ABCBA ABA

A

#### Write a program to display M pattern with stars?

class DisplayM

{

public static void main(String[] args)

{

int spaces=8;

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

{

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

{

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

}

for ( int k=1;k<=spaces ; k++)

{

System.out.print(" ");

}

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

{

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

}

}

###### OUTPUT:

\* \*

\*\* \*\*

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

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

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

}

System.out.println(); spaces -=2;

}

#### Write a program to display sequence of numbers in TRIANGLE format?

import java.util.Scanner; class Series

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter the rows"); int n = sc.nextInt();

int k =0;

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

{

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

{

k++;

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

}

System.out.println(" ");

}

}

}

###### OUTPUT:

enter the rows: 5 1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

***Programs on***

***Strings***

#### Write a program to find weather a string is ANAGRAM or not?

**Def:** a word, phrase, or name formed by rearranging the letters of another, such as *silent* formed from

*listen*.

class Anagram

{

static String removeSpaces(String str)

{

char [] ch=str.toCharArray ();

//convert the string into array String nstr=" ";

//create a new empty string

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

{

if(ch[i]!=' ')

nstr=nstr + ch[i];

/\* if the character at ith index is not equal to space then add that character to new empty string\*/

}

return nstr;

}

static String toLowerCase(String str)

{

char[] ch=str.toCharArray();

//convert the string into array String nstr=" ";

//create a new empty string

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

{

if(ch[i]>=65 && ch[i]<=90)

{

}

else

{

nstr=nstr+((char)ch[i]+32);

/\*if any alphabet is in upper case convert it into lower case\*/

nstr=nstr + ch[i];

//if it is in lower case no need to convert

}

}

return nstr;

}

static String sort(String str)

{

char[] ch=str.toCharArray();

//sort string in alphabetical order for(int i=0;i<ch.length-1;i++)

{

for(int j=i+1;j<ch.length;j++)

{

if(ch[i]>ch[j])

{

char t=ch[i]; ch[i]=ch[j]; ch[j]=t;

}

}

}

String st=new String(ch); return st;

}

static boolean compare(String s1, String s2)

{

if(s1.length()!=s2.length()) return false;

else

{

s1=toLowerCase(s1); s2=toLowerCase(s2); s1=sort(s1); s2=sort(s2);

char ch1[]=s1.toCharArray(); char ch2[]=s2.toCharArray();

for(int i=0;i<ch1.length;i++)

{

if (ch1[i]!=ch2[i])

{

return false;

}

}

return true;

}

}

public static void main(String[] args)

{

java.util.Scanner sc=new java.util.Scanner(System.in); System.out.println ("Enter the first string");

String s1=sc.nextLine();

System.out.println ("Enter the second string"); String s2=sc.nextLine();

s1=removeSpaces (s1); s2=removeSpaces (s2); boolean b= compare(s1,s2);

if(b)

else

}

}

System.out.println("string is anagram"); System.out.println("not an anagram");

###### Output:

Enter the first string Mother in law

Enter the second string Hitler woman

string is anagram

**Write program weather the string is PANAGRAM or not? Def:** a sentence containing every letter of the alphabet.

import java.util.Scanner; public class Panagram

{

public static void main(String[] args) {

Scanner sc = new Scanner(System.in); System.out.println("enter the string "); String s = sc.nextLine();

System.out.println("given string is :"+"\n" +s); String st=removeSpace(s);

int d = check(st); if(d == -1)

System.out.print(s+"\n" + "is not pangram"); else

System.out.print(s+"\n" +"is a pangram");

}

public static String removeSpace(String s)

{

char ch[]=s.toCharArray(); String nstr="";

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

{

if (ch[i]!=' ')

{

}

return nstr;

}

nstr=nstr + ch[i];

}

public static int check(String st)

{

int n = 26;

/\*if(s.length() < n){

return -1; use these lines only for perfect Panagram i.e., it must contain only

}\*/ 26 letters (alphabets) without any repetition.

for(char i = 'A'; i <= 'Z' ; i++){

if((st.indexOf(i) < 0) && (st.indexOf((char)(i + 32)) < 0))

{

return -1;

}

}

return 1;

}

}

###### OUTPUT:

enter the string:

the quick brown fox jumps over a lazy dog given string is :

the quick brown fox jumps over a lazy dog the quick brown fox jumps over a lazy dog is a pangram

#### Write a program check the given string is PALINDROME or not?

import java.util.Scanner; public class PalindromeStr

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter the string"); String st=sc.nextLine();

String nstr="";

char ch[]=st .toCharArray(); for (int i=0 ;i<ch.length/2;i++ )

{

char t=ch[i]; ch[i]=ch[ch.length-1-i]; ch[ch.length-1-i]=t;

}

nstr=new String (ch);

if(nstr.equalsIgnoreCase(st)) System.out.println( st+" string is palindrome "); else

System.out.println(st+" string is not palindrome");

}

}

###### OUTPUT:

Enter the string: Malayalam Malayalam string is palindrome

#### Write a program to display REVERSE of a STRING?

import java.util.Scanner; class Revstring

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter the string"); String st=sc.nextLine();

char ch[]=st.toCharArray(); for (int i=0 ;i<ch.length/2;i++ )

{

char t=ch[i]; ch[i]=ch[ch.length-1-i]; ch[ch.length-1-i]=t;

}

st=new String (ch);

System.out.println("Reserved string is :"+st);

}

}

###### OUTPUT:

enter the string rama and laxmana

Reserved string is : anamxal dna amar

#### Write a program to COUNT number of CHARACTERS in a String?

import java.util.Scanner;

public class NoOfCharactersInaString

{

public static void main(String[] args)

{

int count=0;

Scanner scn=new Scanner(System.in); System.out.println("Enter a string ");

String st=scn.nextLine(); char ch[]=st.toCharArray();

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

{

if(ch[i]>=65&&ch[i]<=90 ||ch[i]>=97 && ch[i]<=122||ch[i]>=48&&ch[i]<=57 &&

ch[i]!=32 && ch[i]!=',' &&ch[i]!='.')

count++;

}

System.out.println("No of Characters="+count);

}

}

###### OUTPUT:

Enter a string:..........

adkvdh dodksk

No of Characters=12

#### Write a program to find the sum of numbers in an ALPHA NUMERIC STRING?

import java.util.Scanner; public class SumOfDigits

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter the alpha numeric string");

String str=sc.nextLine(); char[] ch=str.toCharArray(); int j=0;

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

{

if(ch[i]>=48 && ch[i]<=57)

{

j+=ch[i]-48;

}

}

System.out.println(j);

}

}

###### OUTPUT:

enter the alpha numeric string 139y1d5801

28

#### Write a Program for number of characters in each WORD and count them?

import java.util.Scanner; class Countword

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter the string"); String s=sc.nextLine();

String nst=" "; int nc=0;

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

{

if (s.charAt(i)==' ')

{

}

else

{

}

}

nst=nst + nc; nc=0;

nc++;

nst=nst + s.charAt(i);

nst=nst + nc;

System.out.println (" no of character in each word in a string is "+ nst);

}

}

###### OUTPUT:

enter the string rama and laxmana

no of character in each word in a string is rama 4 and 3 laxmana 7

#### Write a Program to display OCCURENCES of each character in a STRING?

import java.util.Scanner; class NumOfOcc

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("Enter the String"); String st = sc.nextLine();

int n=st.length();

char ch[]=st.toCharArray(); for (int i=0;i<n ;i++ )

{

int count=1;

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

{

if(ch[i]==ch[j])

{

count++; int k=j;

while (k<n-1)

{

}

n--;

j--;

}

}

ch[k]=ch[k+1]; k++;

System.out.println(ch[i]+" occurred "+count+" times");

}

String nst=" ";

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

{

nst=nst + ch[i];

}

System.out.println(nst);

}

}

###### OUTPUT:

Enter the String Malayalam m occurred 2 times

a occurred 4 times l occurred 2 times y occurred 1 times maly

#### Write a program to display number of LOWERCASE, UPPERCASE, SPECIAL SYMBOLS, SPACES and DIGITS in a STRING?

import java.util.Scanner; class DiffTypeCharsSymbols

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter the string"); String st=sc.nextLine();

char ch[]=st.toCharArray();

int uc=0,lc=0,spc=0,dc=0,sp=0; for (int i=0;i<ch.length ;i++ )

{

if (ch[i]>=65&&ch[i]<=90)

{ uc++;

}

else if (ch[i]>=97&&ch[i]<=122)

{

lc++;

}

else if (ch[i]>=48&&ch[i]<=57) dc++;

else

if(ch[i]==' ') sp++;

else spc++;

}

System.out.println("no :of upper case letter "+uc); System.out.println("no: of lower case letter" +lc); System.out.println("no: of decimal number" +dc); System.out.println("no: of spaces "+sp); System.out.println("no: of special characters" +spc);

}

}

###### OUTPUT:

enter the string: [PramoD123$@gmail.com](mailto:PramoD123%24@gmail.com) no :of upper case letter 2

no : of lower case letter12 no : of decimal number3 no : of spaces 0

no : of special characters3

#### Write a program to convert NUMBER into WORDS?

import java.util.\*;

public class Numtoword

{

static String one[]={"","one","two","three","four","five","six","seven","eight","nine","ten",

"eleven","tweleve","thirteen","fourteen","fifteeen","sixteeen","seventeen","eighteen","nineteen"}; static String two[]={"","","twenty","thirty","fourty","fifty","sixty","seventy","eigty","ninety"};

static void pw(int n, String st)

{

if(n<=19)

System.out.print(one[n]+" ");

else

System.out.print(two[n/10]+one[n%10]+" ");

if(n!=0)

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

}

public static void main(String[] args)

{

System.out.println("enter the number"); Scanner sc=new Scanner(System.in);

int num=sc.nextInt(); pw(num/10000000,"crores"); pw((num/100000)%100,"Lakhs"); pw((num/1000)%100,"Thousand"); pw((num/100)%10,"Hundered"); pw(num%100," ");

}

}

###### OUTPUT:

enter the number : 999999

nine Lakhs ninety nine Thousand nine Hundered and ninety nine

#### Write a program to REVERSE the SENTENCE?

import java.util.Scanner; class Revsentence

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter the sentence"); String st=sc.nextLine();

char ch[]=st.toCharArray(); String rst=" ";

for (int i=ch.length-1;i>=0 ;i-- )

{

int k=i;

while (i>=0&&ch [i]!=' ')

{

i--;

}

int j=i+1; while ( j<=k)

{

rst =rst +ch[j]; j++;

}

rst=rst+' ';

}

System.out.println("The reserve sentence is:"+rst);

}

}

###### OUTPUT:

enter the sentence: rama and laxmana

The reserve sentence is: laxmana and rama

#### Write a program to REVERSE THE WORDS in a SENTENCE?

import java.util.Scanner; class Revwords

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter the sentence"); String st=sc.nextLine();

char ch[]=st.toCharArray(); String rst=" ";

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

{

int k=i;

while (i<ch.length &&ch [i]!=' ')

{

i++;

}

int j=i-1; while ( k<=j)

{

rst=rst + ch[j]; j--;

}

rst=rst+' ';

}

System.out.println("The reserved words of sentence is:"+rst);

}

}

###### OUTPUT:

enter the sentence: **rama and laxmana**

The reserved words of sentence is: **amar dna anamxal**

#### Write a program to display STRING INITCAP of Words?

import java.util.Scanner; class Stringinitcap

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter the string"); String st=sc.nextLine();

char ch[]=st.toCharArray(); for (int i=0 ;i<ch.length;i++ )

{

if (i==0||(ch[i]!=' '&&ch[i-1]==' '))

{

if (ch[i]>=97&&ch[i]<=122)

{

ch[i]=(char)(ch[i]-32);

}

else if (ch[i]>=65&&ch[i]<=90)

{

ch[i]=(char)(ch[i]-32);

}

}

}

st=new String(ch);

System.out.println("enter the string in it cap : "+st);

}

}

###### OUTPUT:

enter the string: pramod reddy pavan chandu

enter the string in it cap : **P**ramod **R**eddy **P**avan **C**handu

#### Write a program to convert UPPER CASE TO LOWER CASE & VICE VERSA?

import java.util.Scanner; class Stringuptolow

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in); System.out.println("enter the string"); String st=sc.nextLine();

char ch[]=st.toCharArray(); for (int i=0 ;i<ch.length;i++ )

{

if (ch[i]>=65&&ch[i]<=90)

{

ch[i]=(char)(ch[i]+32);

}

else if (ch[i]>=97&&ch[i]<=122)

{

ch[i]=(char)(ch[i]-32);

}

}

st=new String(ch); System.out.println("converted String in Case : "+st);

}

}

###### OUTPUT:

enter the string : PraMoD ReddY GoPi RedDY converted String in Case : pRAmOd rEDDy gOpI rEDdy

#### Write a program to find a SUB-STRING without using INBUILT functions?

import java.util.Scanner; class Substring

{

public static void main(String[] args)

{

System.out.println("enter the main string"); Scanner sc=new Scanner(System.in);

String st1=sc.next();

char ch1[]=st1.toCharArray(); System.out.println("enter the sub string"); String st2=sc.next();

char ch2[]=st2.toCharArray(); int find=0;

for (int i=0;i<ch1.length ;i++ )

{

int k=i, j=0;

while (k<ch1.length && j<ch2.length && ch1[k]==ch2[j])

{

j++; k++;

}

if(j==ch2.length)

{

"+k+" indexs");

}

find++;

System.out.println( find+" times "+st2+" present between "+i+" to

}

if(find==0)

System.out.println("not found");

}

}

**OUTPUT:**

enter the main string : PramodReddy enter the sub string : Reddy

1 times Reddy present between 6 to 11 indexs

#### Write a program to convert Integer of String type to INTEGER type without using parse int?

import java.util.Scanner; public class StringToInt

{

public static void main (String [] args)

{

Scanner sc=new Scanner (System.in); System.out.println ("enter the String"); String s=sc.next ();

System.out.println (" After converting string to integer"); int d = check(s);

if (d==0)

System.out.println ("not valid string ");

else

System.out.println (d + “is in integer type");

}

public static int check (String s)

{

int i=0, number=0;

for (int j = 0; j < s.length (); j++)

{

char ch [] =s.toCharArray ();

if (ch[j]>'a'&&ch[j] <='z'||ch[j]>'A'&&ch[j]<='Z')

{

return 0;

}

}

while (i<s.length ())

{

number= number\*10; number=number+ (s.charAt (i++)-'0');

}

return number;

}

}

###### OUTPUT:

enter the String 3306

After converting string to integer 3306 is in integer type

## SEARCHING & SORTING PROGRAMS

#### Write a program for LINEAR SEARCH?

public class SearchLinear

{

public static int linearSearch(int[] arr, int x)

{

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

{

if(x==arr[i])

{

return i;

}

}

return -1;

}

public static void main(String[] args)

{

int[] ar ={3,46,76,4,89,7,27};

System.out.println(linearSearch(ar,4)); System.out.println(linearSearch(ar,78));

}

}

###### OUTPUT:

3

-1

#### Write a program for BINARY SEARCH?

public class SearchBinary

{

public static int binarySearch(int[] arr, int x)

{

int first=0;

int last=arr.length-1; while(first<=l)

{

int middle=(first + last)/2; if(x==arr[middle])

{

return middle;

}

else if(x>arr[middle])

{

}

else

{

}

}return -1;

}

first=middle+1;

last=middle-1;

public static void main(String[] args)

{

}

} **OUTPUT:** 1

int[] i={10,49,67,90,40,86};

System.out.println(binarySearch(i,49));

#### Write a program for BUBBLE SORT?

class Bubbledown

{

public static void sortdown(int[]a)

{

int n=a.length;

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

{

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

{

if(a[i]>a[j])

{

int temp=a[i]; a[i]=a[j]; a[j]=temp;

}

}

}

}

public static void main(String[] args)

{

int []a={5,8,1,6,9,2};

sortdown(a); for (int x: a )

{

System.out.println(x);

}

}

}

###### OUTPUT:

1

2

5

6

8

9

# PROGRAMS on

**ARRAYS**

#### Write a program to INSERT the ELEMENTS in an Array?

import java.util.Scanner; public class InstSingArray

{

public static void main (String [ ] args)

{

Scanner sc= new Scanner (System.in); System.out.println ("enter the size"); int length= sc.nextInt ();

int arr [ ] =new int [length];

System.out.println ("enter the "+length+" elements"); for (int i = 0; i < arr.length; i++)

{

arr[i] =sc.nextInt ();

}

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

{

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

}

}

}

**Output:** enter the size 5

Enter the 5 elements 2

3

5

8

64

arr [0] >2

arr [1] >3

arr [2] >5

arr [3] >8

arr [4] >64

#### Write a Program to REVERSE THE ELEMENTS of an array?

import java.util.Scanner; public class InstSingArray

{

public static void main (String [ ] args)

{

Scanner sc= new Scanner (System.in); System.out.println ("enter the size"); int length= sc.nextInt ();

int arr [ ] =new int [length];

System.out.println ("enter the "+length+" elements"); for (int i = 0; i < arr.length; i++)

{

arr[i] =sc.nextInt ();

}

System.out.println (“Before Reverse of an Array”); for (int i = 0; i < arr.length; i++)

{

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

}

for (int i = 0; i < arr.length/2; i++)

{

int t=arr[i];

arr[i] =arr [arr.length-1-i]; arr [arr.length-1-i) =t;

}

System.out.println (“After Reverse of an Array”); for (int i = 0; i < arr.length; i++)

{

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

}

}

}

###### Output:

Enter the size 5

Enter the 5 elements 1

5

6

8

9

Before Reverse of an Array arr [0] >1

arr [1] >5

arr [2] >6

arr [3] >8

arr [4] >9

After Reverse of an Array arr [0] 9

arr [1] 8

arr [2] 6

arr [3] 5

arr [4] 1

#### Write a program to INSERT A ELEMENT INTO EXISTING ARRAY in a specified position?

**import** java.util.Scanner;

**class** Insertingelement

{

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

{

Scanner sc= **new** Scanner (System.***in***); System.***out***.println ("enter the length"); **int** length= sc.nextInt ();

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

System.***out***.println ("enter the "+length+" elements");

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

{

arr[i]=sc.nextInt();

}

System.***out***.println ("length of array before inserting"+"--->"+arr.length);

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

{

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

}

System.***out***.println ("enter the index of specified position or index");

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

System.***out***.println ("enter the element to replace to specified index");

**int** ele=sc.nextInt (); arr=*insert* (arr ,in ,ele);

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

{

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

}

}

**static int** [] insert (**int** a[],**int** in, **int** ele)

{

i**f** (in>a.length||in<0)

{

System.***out***.println ("invalid index");

**return** a;

}

###### else

{

**int** na [] = **new int** [a.length+1];

**for** (**int** i= 0 ; i<in ;i++ )

{

na[i] = a[i];

}

na [in] =ele;

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

{

na [i+1] = a[i];

}

System.***out***.println ("length of array after inserting"+"--->"+na.length);

**return** na;

}

}

}

###### Output:

enter the length 5

enter the 5 elements 2

8

6

7

88

length of array before inserting--->5 0 >2

1------->8

2------->6

3------->7

4------->88

enter the index of specified position or index 3

enter the element to replace to specified index 62

length of array after inserting >6

0------->2

1------->8

2------->6

3------->62

4------->7

5------->88

#### Write a program to DELETE AN ELEMENT OF A SPECIFIED INDEX IN THE EXISTING ARRAY?

import java.util.Scanner; class DeletingArray

{

public static void main (String [] args)

{

Scanner sc= new Scanner (System.*in*); System.*out*.println ("enter the length"); int length= sc.nextInt ();

int ar [] = new int [length];

System.*out*.println ("enter the "+length+" elements"); for (int i = 0; i < ar.length; i++)

{

ar[i] = sc.nextInt ();

}

System.*out*.println ("length of array before deleting"+"--->"+ar.length);

*display* (ar);

System.*out*.println ("enter specified position for deleting that element"); int in=sc.nextInt ();

ar=*delete* (ar , in);

*display* (ar);

}

static void display (int a[])

{

for (int i=0; i<a.length; i++) System.*out*.println (i+" >"+a[i]);

}

static int [] delete (int a[] , int in)

{

If (in>a.length||in<0)

{

System.*out*.println ("invalid index"); return a;

}

else

{

int na [] = new int [a.length-1]; for (int i=0; i<in; i++)

{

na[i] = a[i];

}

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

{

na [i-1] = a[i];

}

System.*out*.println ("length of array after deleting"+" >"+na.length);

return na;

}

}

}

###### OUTPUT:

enter the length 6

enter the 6 elements 5

5

9

8

6

2

length of array before deleting--->6 0 >5

1------->5

2------->9

3------->8

4------->6

5------->2

enter specified position for deleting that element 5

length of array after deleting >5

0------->5

1------->5

2------->9

3------->8

4------->2

#### Write a program to SEARCH AN ELEMENT IN THE EXISTING ARRAY?

public class Search element

{

public static void main (String [] args)

{

int ar [] = {22, 11, 23, 11, 15, 19};

int inx=search (ar, 15); display (ar);

if (inx>=0)

System.out.println ("your element found at index "+inx);

else

System.out.println ("not valid");

}

static void display (int a [])

{

for (int i=0; i<a.length; i++) System.out.println (i+" >"+a[i]);

}

static int search (int a [], int ele)

{

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

{

If (ele==a[i])

return i;

}

return -1;

}

}

###### OUTPUT:

0------->22

1------->11

2------->23

3------->11

4------->15

5------->19

your element found at index 4

#### Write a program to find BIGGEST AND SMALLEST ELEMENT in the given array?

import java.util.Scanner; public class BigeleArray

{

public static void main (String [] args)

{

Scanner sc= new Scanner (System.in); System.out.println ("enter the length"); int length= sc.nextInt ();

int arr [] =new int [length]; int bigger=0;

int smaller = 0;

System.out.println ("enter the "+length+ “elements"); for (int i = 0; i < arr.length; i++)

{

arr[i] = sc.nextInt ();

}

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

{

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

}

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

{

int big=arr [0];

int small=arr [0]; if (big<arr[i])

{

big=arr[i];

}

if (small>arr[i])

{

small=arr[i];

}

smaller=small; bigger=big;

}

System.out.println ("biggest element is >"+bigger);

System.out.println ("Smallest element is >"+smaller);

}

}

###### OUTPUT:

enter the length 5

enter the 5elements 1

8

99

66

75

arr [0] >1

arr [1] >8

arr [2] >99

arr [3] >66

arr [4] >75

biggest element is >75

Smallest element is >1

#### Write a program to find FIRST BIGGEST AND SECOND BIGGEST ELEMENT in given array?

class Fbiggest

{

public static void main (String [] args)

{

int ar[]={23,12,14,56,22,28,13};

int fbig=ar [0];

int sbig=ar [1];

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

{

if (fbig<ar[i])

{

sbig=fbig; fbig=ar[i];

}

else if (sbig<ar[i])

{

sbig=ar[i];

}

}

System.out.println ("first biggest element is “+fbig); System.out.println ("second biggest element is “+sbig);

}

}

###### OUTPUT:

First biggest element is 56 Second biggest element is 28

#### Write a program to FIND THE SECOND OCCURRENCE ELEMENT in a given array?

class Secondoccuranceelement

{

public static void main (String [] args)

{

int ar[]={22,11,23,11,15,19,11};

int inx=secondoccurance (ar, 11); display (ar);

if (inx>=0)

System.out.println ("Second time occurred element found at the index "+inx);

else

System.out.println ("not valid");

}

static void display (int a [])

{

for (int i=0; i<a.length; i++) System.out.println ("arr ["+i+"]"+" >"+a[i]);

}

static int secondoccurance (int a [], int ele)

{

int count=0;

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

{

If (ele==a[i])

count++; if (count==2)

return i;

}

return -1;

}

}

###### OUTPUT:

arr [0] >22

arr [1] >11

arr [2] >23

arr [3] >11

arr [4] >15

arr [5] >19

arr [6] >11

Second time occurred element found at the index 3

#### Write a program to FIND THE OCCURRENCE ELEMENT IN which position in a given array?

class Occuranceelement

{

public static void main (String [] args)

{

int ar[]={22,11,23,11,15,19,11};

int inx=occurrence (ar, 11, 2); display (ar);

if (inx>=0)

System.out.println ("your element found at index "+inx);

else

System.out.println ("not valid");

}

static void display (int a [])

{

for (int i=0; i<a.length; i++) System.out.println (i+" >"+a[i]);

}

static int occurrence (int a [], int ele, int oc)

{

int count=0;

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

{

if (ele==a[i])

count++; if (count==oc)

return i;

}

return -1;

}

}

###### OUTPUT:

0------->22

1------->11

2------->23

3------->11

4------->15

5------->19

6------->11

Your element found at index 3

#### Write a program to FIND HOW MANY TIMES ELEMENT IS OCCURED in a given array?

class Elementoccured

{

public static void main (String [] args)

{

int ar[]={22,11,23,11,15,19,11};

int in=occurred (ar, 11); display (ar);

if (in>=0)

System.out.println ("your element occurred “+in);

else

System.out.println ("not valid");

}

static void display (int a [])

{

for (int i=0; i<a.length; i++) System.out.println (i+" >"+a[i]);

}

static int occurred (int a [], int ele)

{

int count=0;

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

{

if (ele==a[i])

count++;

}

return count;

}

}

###### OUTPUT:

0------->22

1------->11

2------->23

3------->11

4------->15

5------->19

6------->11

Your element occurred 3

#### Write a program to DISPLAY MISSING ELEMENT in a given sorted array?

class Missingelement

{

public static void main (String [] args)

{

int ar [] = {8, 15, 21, 24, 30, 37};

System.out.println ("Missing elements in given array are :"); for (int i=0;i<ar.length-1 ;i++ )

{

for (int j=ar[i]+1;j<ar[i+1]; j++ )

{

System.out.println (j);

}

}

}

}

###### OUTPUT:

Missing elements in given array are:

9 ,10 ,11 ,12 ,13 ,14 ,16 ,17 ,18 ,19 ,20 ,22 ,23 ,25 ,26 ,27 ,28 ,29 ,31 ,32 ,33 ,34 ,35 ,36

#### Write a program to FIND HIGHEST CONTIGUOUS SUM OF TWO ELEMENT in a given array?

public class Sumoftwoelemnts

{

public static void main (String [] args)

{

int ar[]={21,12,15,32,16,17,22};

int inx=0;

int big=ar [0] +ar [1];

for (int i=1; i<ar.length-1; i++)

{

if (big<ar[i] + ar [i+1])

{

big=ar[i] +ar [i+1]; inx=i;

}

}

System.out.println (“sum of two element"+" >"+big);

System.out.println (“the first element"+" >"+ar [inx]);

System.out.println (" the second element"+" >"+ar [inx+1]);

}

}

###### OUTPUT:

Sum of two element >48

The first element >32

The second element >16

#### Write a program to DISPLAY THE COMMON ELEMENTS between two arrays?

public class Commonelement

{

public static void main (String [] args)

{

int ar1 [] = {12, 13, 23, 15, 11, 16};

int ar2 [] = {53, 26, 23, 15, 18, 13};

System.out.println ("common elements are: "); for (int i=0; i<ar1.length; i++)

{

for (int j=0; j<ar2.length;j++ )

{

if (ar1 [i] ==ar2 [j])

{

System.out.println (ar1 [i]);

break;

}

}

}

}

}

###### OUTPUT:

common elements are:

13

23

15

#### Write a program to EXCHANGE OF FIRST PART ELEMENT TO SECOND PART

**Element between two arrays?**

public class Exchangeofelements

{

public static void main (String [] args)

{

int ar[]={21,12,15,32,16,17,22};

System.out.println ("BEFORE EXCHANGE OF ARRAY"); for (int i = 0; i < ar.length; i++)

{

System.out.println (ar[i]);

}

int n;

if (ar.length%2==0)

n=ar.length/2;

else

n= (ar.length/2) +1;

for (int i=0; i<ar.length/2; i++)

{

int t=ar[i]; ar[i] =ar [n+i]; ar [n+i]=t;

}

System.out.println ("AFTER EXCHANGE OF ARRAY"); for (int i = 0; i < ar.length; i++)

{

System.out.println (ar[i]);

}

}

}

###### OUTPUT:

BEFORE EXCHANGE OF ARRAY 21

12

15

32

16

17

22

AFTER EXCHANGE OF ARRAY 16

17

22

32

21

12

15

#### Write program TO DISPLAY DISTINCT ELEMENTS from given two array?

public class Disctinctelements

{

public static void main (String [] args)

{

int ar1 [] = {12, 13, 23, 15, 11, 16};

int ar2 [] = {53, 26, 23, 15, 18, 13};

System.out.println ("Distinct elements from given two arrays"); for (int i=0; i<ar1.length; i++)

{

int find=0;

for (int j=0; j<ar2.length; j++)

{

if (ar1 [i] ==ar2 [j])

{

find=1; break;

}

}

if (find==0)

}

System.out.println (ar1 [i]);

for (int i=0; i<ar2.length; i++)

{int find=0;

for (int j=0; j<ar1.length; j++)

{

if (ar2 [i] ==ar1 [j])

{

find=1; break;

}

}

if (find==0)

}

}

}

System.out.println (ar2 [i]);

###### OUTPUT:

Distinct elements from given two arrays 12

11

16

53

26

18

13

13

#### Write a program to MERGE TWO ARRAYS?

public class Merge

{

public static void main (String [] args)

{

int ar1 [] = {12, 13, 23, 15, 11, 16};

int ar2 [] = {53, 26, 23, 15, 18, 13};

int res [] =new int [ar1.length+ar2.length]; int j=0;

for (int i = 0; i < ar1.length; i++, j++)

{

res[j] =ar1 [i];

}

for (int i = 0; i < ar2.length; i++, j++)

{

res[j] =ar2 [i];

}

System.out.println ("MERGED ARRAY "); for (int i = 0; i < res.length; i++)

{

System.out.println (res[i]);

}

}

}

###### OUTPUT:

MERGED ARRAY 12

13

23

15

11

16

53

26

23

15

18

13

#### Write a program to COMBINE TWO ARRAYS IN ZIGZAG manner?

public class Zigzag

{

public static void main (String [] args)

{

int ar1 [] = {12, 13, 23, 15, 11, 16};

int ar2[]={53,26,23,15,18,13,23,45};

int res [] =new int [ar1.length+ar2.length]; int i=0, j=0;

for (int k = 0; k < res.length; )

{

if (i<ar1.length)

{

res[k] =ar1 [i]; i++;

k++;

}

if (j<ar2.length)

{

res[k] =ar2 [j]; j++;

k++;

}

}

System.*out*.println ("ZIGZAG ARRAY IS"); for (int l = 0; l < res.length; l++)

{

System.*out*.println (res[l]);

}

}

}

###### OUTPUT:

ZIGZAG ARRAY IS 12

53

13

26

23

23

15

15

11

18

16

13

23

45

#### Write a program to find the PALINDROME numbers in the given ARRAY?

class Main3

{

static void display (int a [])

{

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

{

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

}

System.out.println ();

}

static int revdig (int n)

{

int rev=0; while (n>0)

{

int r=n%10; rev=rev\*10+r; n=n/10;

}

return rev;

}

public static void main (String [] args)

{

int ar [] = {232, 12, 78, 898, 34543, 45};

display (ar); int count=0;

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

{

if (ar [i] ==revdig (ar[i])) count++;

}

System.out.println (" ");

System.out.println (" number of palindrome:"+count);

}

}

###### OUTPUT:

232, 12,78,898,34543,45,

---------------------

number of palindrome: 3

#### Write a program to read elements into the MATRIX from SCANNER?

import java.util.\*; class Main2

{

static int [] [] readMat ()

{

Scanner sc= new Scanner (System.in); System.out.println ("Enter the Order");

int m=sc.nextInt ();

int n=sc.nextInt ();

int ar [][] =new int[m][n];

System.out.println ("enter "+m\*n+" Elements"); for (int i=0; i<ar.length; i++)

{

for (int j=0; j<ar[i].length; j++)

{

ar[i] [j] =sc.nextInt ();

}

}

return ar;

}

static void display (int a [] [])

{

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

{

for (int j=0; j<a[i].length; j++)

{

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

}

System.out.println ();

}

}

public static void main (String [] args)

{

int ar [] []=readMat(); System.out.println ("Entered Matrix :"); display (ar);

}

}

###### OUTPUT:

Enter the Order 2

2

enter 4 Elements

9

6

5

1

Entered Matrix:

9 6

5 1

#### Write a program to read inputs from SCANNER and find the BIGGEST ELEMENT in EACH ROW and EACH COLUMN?

import java .util.\*; class Readmatrix

{

public static void main (String [] args)

{

Scanner sc=new Scanner (System.in);

System.out.println ("enter the order"); int m=sc.nextInt ();

int n=sc.nextInt ();

int ar[][]=new int [m][n];

System.out.println ("enter" + m\*n + " elements"); for (int i=0;i<ar.length ;i++ )

{

for (int j=0;j<ar[i].length ;j++ )

{

ar[i][j]=sc.nextInt();

}

}

System.out.println (" entered matrix:"); for (int i=0;i<ar.length ;i++ )

{

for (int j=0; j<ar[i].length ; j++ )

{

System.out.print (ar[i][j]+"("+i+","+j+")");

}

System.out.println ();

}

System.out.println ();

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

{

int big=ar[i][0];

for (int j=i ; j<ar[i].length ;j++)

{

if(big<ar[i][j])

big = ar[i][j]; break;

}

System.out.println (i+1+"row biggest element "+big);

}

for (int i=0; i<ar[0].length ;i++ )

{

int big=ar[0][i];

for (int j=0;j<ar.length ;j++ )

{

if (big<ar[j][i])

big =ar[j][i];

}

System.out.println(i+1+"column biggest element "+big);

}

}

}

###### OUTPUT:

enter the order 2

2

enter4elements 5

6

8

9

entered matrix:

5(0,0)6(0,1)

8(1,0)9(1,1)

1row biggest element5 2row biggest element9 1column biggest element8 2column biggest element9

#### Write a program to read inputs from SCANNER and find the SUM of ELEMENTS in EACH ROW and EACH COLUMN?

import java.util.\*;

class Rowwiseandcolwisesum

{

static int [][] readMat()

{

Scanner sc=new Scanner(System.in); System.out.println("enter the order");

int m=sc.nextInt(); int n=sc.nextInt();

int ar[][]=new int [m][n]; System.out.println("enter"+ m\*n+ "elements");

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

{

for (int j=0;j<ar[i].length ;j++ )

{

ar[i][j]=sc.nextInt();

}

}

return ar;

}

static void display(int a[][])

{

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

{

for (int j=0;j<a[i].length ;j++ )

{

System.out.print(a[i][j]+"("+i+","+j+")");

}

System.out.println();

}

}

public static void main(String[] args)

{

int ar[][]=readMat(); System.out.println("entered matrix"); display(ar);

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

{

int rsum=0; int csum=0;

for (int j=0;j<ar.length ;j++)

{

rsum=rsum + ar[i][j]; csum=csum + ar[j][i];

}

System.out.println(i+1+"row sum is :"+rsum); System.out.println(i+1+"column sum is:"+csum);

}

}

}

###### OUTPUT:

enter the order 2

2

enter4elements 6

5

7

9

entered matrix 6(0,0)5(0,1)

7(1,0)9(1,1)

1row sum is :11 1column sum is:13 2row sum is :16 2column sum is: 14

# SPECIAL PROGRAMS

#### Write a program to find the given YEAR is LEAP-YEAR or not?

import java.util.\*; public class Leapyear

{

public static void main (String [] args)

{

Scanner sc=new Scanner (System.in); System.out.println ("Enter the year") ; int m=sc.nextInt ();

if (m%4==0&&m%100! =0||m%400==0) System.out.println ("it is a leap year");

else

}

}

System.out.println ("not a leap year");

###### OUTPUT:

Enter the year 1990

not a leap year

Enter the year 2016

it is a leap year

#### Write a program to find days between DATE to DATE?

import java.util.Scanner; class Date

{

final int m[]={31,28,31,30,31,30,31,31,30,31,30,31};

int dd, mm, yyyy;

Date (int dd, int mm, int yyyy)

{

this.dd=dd; this.mm=mm; this.yyyy=yyyy;

}

int getNumberOfLeapYear ()

{

if (mm>2)

return yyyy/4-yyyy/100+yyyy/400;

else

}

return (yyyy-1)/4-(yyyy-1)/100+ (yyyy-1)/400;

int getNumberOfDays ()

{

int dCount= yyyy\*365+getNumberOfLeapYear () +dd; for (int i=0; i<mm-1; i++)

{

dCount+=m[i];

}

return dCount;

}

int difference (Date d1, Date d2)

{

int dy1=d1. getNumberOfDays (); int dy2=d2. getNumberOfDays (); if (dy1>dy2)

return dy1-dy2;

else

}

return dy2-dy1;

public String toString ()

{

return dd+":"+mm+":"+yyyy+" ";

}

static Date readDate ()

{

Scanner sc= new Scanner (System.in); System.out.println ("Enter dd: ");

int dd=sc.nextInt (); System.out.println ("Enter mm: "); int mm=sc.nextInt (); System.out.println ("Enter yyyy: "); int yy=sc.nextInt ();

return new Date (dd, mm, yyyy);

}

public static void main (String [] args)

{

Date date1=readDate (); Date date2=readDate ();

System.out.println ("Number of Days between"+date1+ "And"+date2+" is: "+date1.difference (date1, date2));

}

}

###### OUTPUT:

Enter dd: 31

Enter mm: 08

Enter yyyy: 2016

Enter dd: 5

Enter mm: 09

Enter yyyy: 2016

Number of Days between31:8:2016 And5:9:2016 is: 5