LOGICAL PROGRAMS

//Write a java program to find the factor of a number

//10--->1,2,5,10 15--->1,3,5,15

class FindFactor

{

public static void main(String[] args)

{

int num=30;

System.out.println("The factor of "+num+" are-:");

for(int i=1;i<=30;i++)//1<=30t 2<=30t 3<=30t

{

if(num%i==0)//30%1==0t 30%2==0t 30%3==0t

System.out.println(i);//1 2 3.....30

}

}

}

//Write a program to find the last digit and first digit of a number

class FirstLastDigit

{

public static void main(String[] args)

{

int num=2345;

int firstDigit=0;

int lastDigit=num%10;//2345%10=5

System.out.println("last Digit="+lastDigit);

while(num!=0)//2345!=0t 234!=0t 23!=0t 2!=0t 0!=0false

{

firstDigit=num%10;//2345%10=5 234%10=4 23%10=3 2%10=2

num=num/10;//2345/10=234 234/10=23 23/10=2 2/10=0

}

System.out.println("First Digit="+firstDigit);//2

}

}

class NeonNumber

{

public static void main(String[] args)

{

int n=9;

int sq=n\*n;//9\*9=81

int ld, sum=0;

while(sq>0)//81>0 t 8>0t 0>0f

{

ld=sq%10;//81%10=1 8%10=8

sum=sum+ld;//0+1=1+8=9

sq=sq/10;//81/10=8 8/10=0

}

if(n==sum)//9==9 t

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

else

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

}

}

class NivenNumber

{

public static void main(String[] args)

{

int num=156,ld,sum=0;

int temp=num;//156-->temp

while(num>0)//156>0t, 15>0t, 1>0t 0>0false

{

ld=num%10;//156%10=6, 15%10=5 1%10=1

sum=sum+ld;//0+6=6+5=11+1=12

num=num/10;//156/10=15, 15/10=1 1/10=0

}

if(temp%sum==0)//156%12==0 true

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

else

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

}

}

class SpyNumber

{

public static void main(String[] args)

{

int num=123,sum=0,lastDigit,prod=1;

while(num>0)//123>0t 12>0t 1>0t 0>0false

{

lastDigit=num%10;//3, 12%10=2, 1%10=1

sum=sum+lastDigit;//0+3=3+2=5+1=6

prod=prod\*lastDigit;//1\*3=3\*2=6\*1=6

num=num/10;//123/10=12, 12/10=1, 1/10=0

}

if(sum==prod)//6==6

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

else

System.out.println("Not a spy number");

}

}

class ReverseNumber

{

public static void main(String[] args)

{

int num=5225,rev=0,lastDigit;

int temp=num;

while(num>0) //1234>0t,, 123>0t,,12>0t,,1>0t 0>0 f

{

lastDigit=num%10;//1234%10=4,,123%10=3,,12%10=2,,1%10=1

rev=rev\*10+lastDigit;//0\*10+4=4 ,,4\*10+3=40+3=43 ,,43\*10+2=430+2=432,, 432\*10+1=4320+1=4321

num=num/10;//1234/10=123,,123/10=12, 12/10=1,, 1/10=0

}

//Palindrome

if(temp==rev)

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

else

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

}

}

class PrimeNumbers

{

public static void main(String[] args)

{

int num=6;

boolean b=false;

for(int i=2;i<=num/2;i++)//2<(20/2)---3t 3<=5t 4<=5t 5<=5t

{

if(num%i==0)//6%2==0 t 11%3==0 f 11%4==0f 11%5==0 f

{

b=true;//true

break;

}

}

if(!b)

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

else

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

}

}

//Write a java program to display the prime number from 1 to n

//divisible by 1 or itself

import java.util.Scanner;

//range of primenumber 1 to n

import java.util.Scanner;

class PrimeNumberRange

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

int count;

System.out.println("Enter the range");

int n=sc.nextInt();

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

for(int i=2;i<=n;i++)//2 3 4

{

count=0;

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

{

if(i%j==0)

{

count++;

}

}

if(count==2)

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

}

}

}

//Write a java program to check that given number is perfect or not

//factor of the number---->add the factors(excluding the number)

//sum of the factor==number(perfect number)

class PerfectNumber //exclude the number itself

{

public static void main(String[] args)

{

int num=12,sum=0;

int i;

for(i=1;i<=num/2;i++)

{

if(num%i==0)

{

sum=sum+i;

}

}

if(sum==num)

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

else

System.out.println("Not a perfect number");

}

}

//Write a Java program to find the factorial of number

class Factorial

{

public static void main(String[] args)

{

int num=7;

int fact=1;

//for(int i=1;i<=num;i++)//1<=5t 2<=5t 3<=5t 4<=5t 5<=5t 6<=5false

for(int i=num;i>=1;i--)//7---1

{

fact=fact\*i;//1\*1=1\*2=2\*3=6\*4=24\*5=120

}

System.out.println("Factorial of "+num+"="+fact);

}

}

//Write a java Program to find the factorial of a given range of numbers

import java.util.Scanner;

class FactorialForARange

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the factorial range number");

int num=sc.nextInt();//4

for(int i=1;i<=num;i++)//1 <=4t 2<=4t 3<=4t 4<=4t 5<=4false

{

System.out.println(i+"!="+fact(i));//1!=1 2!=2 3!=6 4!=24

}

}

public static int fact(int n)//1 2 3 4

{

int fact=1;

while(n>0)//1>0t 0>0f 2>0t 1>0t 0>0f 3>0t 2>0t 1>0t 0>0false 4>0t 3>0t 2>0t 1>0t 0>0f

{

fact=fact\*n;//1\*1=1 1\*2=2\*1=2 1\*3=3\*2=6\*1=6 1\*4=4\*3=12\*2=24\*1=24

n--;//1--=0 2--=1--=0 3--=2--=1--=0 4--=3--=2--=1--=0

}

return fact;

}

}

//Write a java program to check if the number is a Strong number

//145--->1!+4!+5!=1+24+120=145--Strong number

//143---->1!+4!+3!=1+24+6=31---->not Strong

//whose sum of factorial of digits is equal to number itself

import java.util.Scanner;

class StrongNumber

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int num=sc.nextInt();//145

int sum=0;

int n=num;//n=145

while(num!=0)//145!=0t 14!=0t 1!=0 0!=0f

{

int rem=num%10;//145%10=5 14%10=4 1%10=1

sum=sum+fact(rem);//fact(5) 0+120=120 fact(4) 120+24=144 fact(1) 144+1=145

num=num/10;//145/10=14 14/10=1 1/10=0

}

if(sum==n)//145==145

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

else

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

}

public static int fact(int n)

{

int fact=1;

while(n>0)//5>0 4>0 1>0

{

fact=fact\*n;//

n--;

}

return fact;//120 24 1

}

}

import java.util.Scanner;

class StrongNumberRange

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt();//1000

System.out.println("Strong Number are");

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

{

int num=i;

int sum=0;

int temp=num;//temp=1

while(num!=0)//

{

int rem=num%10;//

sum=sum+fact(rem);//

num=num/10;//

}

if(sum==temp)

System.out.println(temp);

}

}

public static int fact(int n)

{

int fact=1;

while(n>0)//5>0 4>0 1>0

{

fact=fact\*n;//

n--;

}

return fact;//120 24 1

}

}

//Fibonacci series

class Fibonacci\_Series

{

public static void main(String[] args)

{

int n1=0,n2=1,n3,count=10;

System.out.print(n1+" "+n2);

for(int i=3;i<=count;i++)

{

n3=n1+n2;

System.out.print(" "+n3);

n1=n2;

n2=n3;

}

}

}

class PrintEvenDigitsFromANumber

{

public static void main(String[] args)

{

int num=254654;

int last, rev=0,rev1,temp=num;

System.out.println("The even number in "+num+"are:-");

for( ; num>0;num/=10)//25654/10=2565>0t

{

last=num%10;//25654%10=4 2565%10=5

rev=rev\*10+last;

}

for( ; rev>0;rev/=10)

{

rev1=rev%10;

if(check(rev1))//4 5

{

System.out.print(rev1);

}

}

}

public static boolean check(int n)//4 5

{

if(n%2==0)//4%2==0t 5%2==0f

return true;

else

return false;

}

}

//25654-->264

//Xylem and Phloem

import java.util.Scanner;

class XylemPhloem

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int num=sc.nextInt();//34326 173156

int sumExtreme=0;//6 9

int sumMean=0;//2 5 9

for(int n=num;n!=0;n=n/10)//n=34326!=0t 34326/10=3432!=0t 3432/10=343!=0t 343/10=34 34/10=3 3/10=0!=0false

{

if(n==num || n<10)//3==34326f 3<10t

sumExtreme+=n%10;//34326%10=6+0=6 3%10=3+6=9

else

sumMean+=n%10;//3432%10=2+0=2 343%10=3+2=5 34%10=4+5=9

}

if(sumExtreme==sumMean)//9==9true

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

else

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

}

}

//Happy Number

class Program16

{

public static void main(String[] args)

{

int num=82;

int res=num;

while(res!=1 && res!=4)

{

res=happyNumber(res);

}

//Happy number always ends with 1

if(res==1)

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

//Unhappy number ends in the repeating cycle of the number which contains 4

else if(res==4)

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

}

public static int happyNumber(int num)

{

int sum=0;

while(num>0)

{

int rem=num%10;

int sq=rem\*rem; //sum=sum+(rem\*rem)

sum=sum+sq;

num=num/10;

}

return sum;

}

}

//Leap year

//concept:- **A year is a leap** year**if it has 366 days in it. According to the proleptic calendar system rules, a year is a Leap year if: If it is divisible by 4. It is not divisible by 100 but it can be divisible by 400.**

**A leap year is exactly divisible by 4 except for century years (years ending with 00). The century year is a leap year only if it is perfectly divisible by 400.**

class Leap\_Year

{

public static void main(String[] args)

{

int year=2024;

if(((year%4==0) && (year%100!=0))||(year%400==0))

System.out.println(year+" is a Leap year");

else

System.out.println(year+" is not a Leap year");

}

}

//Armstrong Number

import java.util.Scanner;

class Program18

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int num=sc.nextInt();//153

if( armstrongNumber(num))//true

System.out.println(num+" is an Armstrong number");

else

System.out.println(num+" is not an Armstrong number");

}

public static boolean armstrongNumber(int n)//153

{

int digits=0,sum=0,temp=n;

while(temp>0)//153>0 t 15>0t 1>0 0>0false

{

temp=temp/10;//153/10=15 15/10=1 1/10=0

digits++;//0++=1++=2++=3

}

temp=n;//temp=153

while(temp>0)//153>0 15>0 1>0 0>0false

{

int rem=temp%10;//153%10=3 15%10=5 1%10=1

sum+=(Math.pow(rem,digits));//Math.pow(3,3)-->27=0+27=27 pow(5,3)--->125=27+125=152

//pow(1,3)-->1=152+1=153

temp=temp/10;//153/10=15 15/10=1 1/10=0

}

if(n==sum)//153==153

return true;

else

return false;

}

}

import java.util.Scanner;

class SumOfEvenAndOddDigitsDigits

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt();

System.out.println("Sum of even digits of "+n+"="+getEven(n));

System.out.println("Sum of odd digits of "+n+"="+getOdd(n));

}

public static int getEven(int num)//2345

{

int evenDigit=0,sum=0;

while(num>0)//2345>0t 234>0 23>0 2>0

{

int rem=num%10;//5 4 3 2

if(rem%2==0)//5%2==0 4%2==0t 3%2==0f 2%2==0

{

evenDigit=evenDigit\*10+rem;//0\*10+4=4\*10+2=42

}

num=num/10;//234 23 2 0

}

while(evenDigit>0)//42>0 4>0t 0>0

{

int rem1=evenDigit%10;//42%10=2 4%10=4

sum=sum+rem1;//0+2=2+4=6

evenDigit=evenDigit/10;//42/10=4/10=0

}

return sum;

}

public static int getOdd(int num)//2345

{

int oddDigit=0,sum=0;

while(num>0)//2345>0t 234>0 23>0 2>0

{

int rem=num%10;//5 4 3 2

if(rem%2!=0)//5%2==0 4%2==0t 3%2==0f 2%2==0

{

oddDigit=oddDigit\*10+rem;//0\*10+4=4\*10+2=42

}

num=num/10;//234 23 2 0

}

while(oddDigit>0)//42>0 4>0t 0>0

{

int rem1=oddDigit%10;//42%10=2 4%10=4

sum=sum+rem1;//0+2=2+4=6

oddDigit=oddDigit/10;//42/10=4/10=0

}

return sum;

}

}

**Recursion:-**

//Reversing a number using recursion

//1234->4321[4 digit] 3->3[1 digit]

**import** java.util.Scanner;

**class** Program14

{

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

{

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

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

**int** n=sc.nextInt();//5432

System.***out***.println("Reverse number");

*rev*(n);//5432 6741

}

**public** **static** **void** rev(**int** n)//5432 543 54 5

{

**if**(n<10)//5432<10f 543<10f 54<10f 5<10t

{

System.***out***.print(n);//5

**return**;

}

**else**

{

System.***out***.print(n%10);//5432%10=2 543%2=3 54%10=4

*rev*(n/10);//5432/10=543->rev(543) 543/10=54->rev(54)

//54/10=5->rev(5)

}

}

}

//factorial with recursion

**class** Program17

{

**static** **int** *fact*=1;

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

{

**int** num=3;//5

*fact*(num);

System.***out***.println("Factorial of "+num+"="+*fact*);

}

**public** **static** **void** fact(**int** n)//3 2 1 0

{

**if**(n>=1)//3>=1 2>=1 1>=1 0>=1f

{

*fact*=*fact*\*n;//1\*3=3\*2=6\*1=6

*fact*(n-1);//fact(2);fact(1);fact(0)

}

}

}

//fibbonaci series using recursion

**class** Program19

{

**static** **int** *n1*=0,*n2*=1,*n3*;

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

{

System.***out***.print(*n1*+" "+*n2*);//0 1 1 2

*fibonacci*(10-2);

}

**public** **static** **void** fibonacci(**int** i)//terms//8 7 6

{

**if**(i>=1)//8>=1t 7>=1t

{

*n3*=*n1*+*n2*;//n3=1 n3=2

System.***out***.print(" "+*n3*);

*n1*=*n2*;//n1=1 1

*n2*=*n3*;//n2=1 2

*fibonacci*(i-1);//8-1 7-1

}

}

}