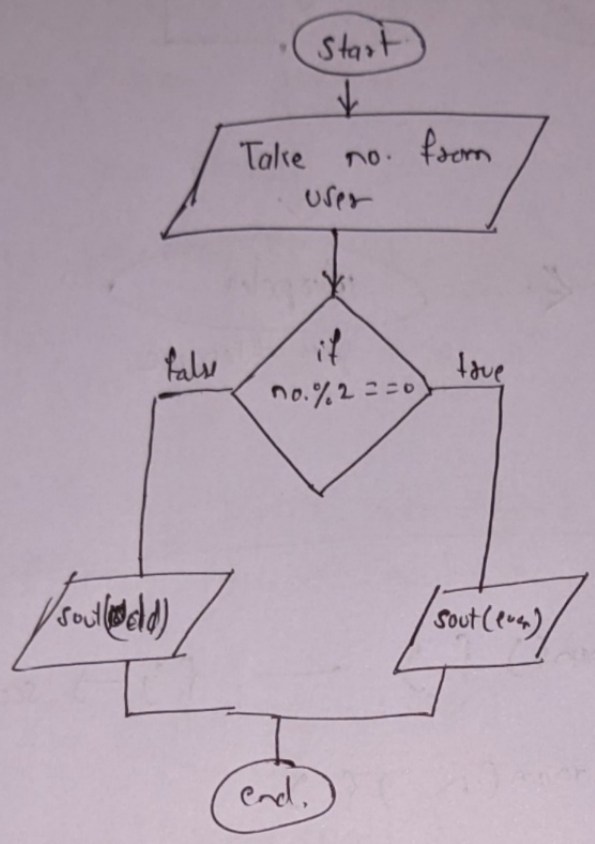
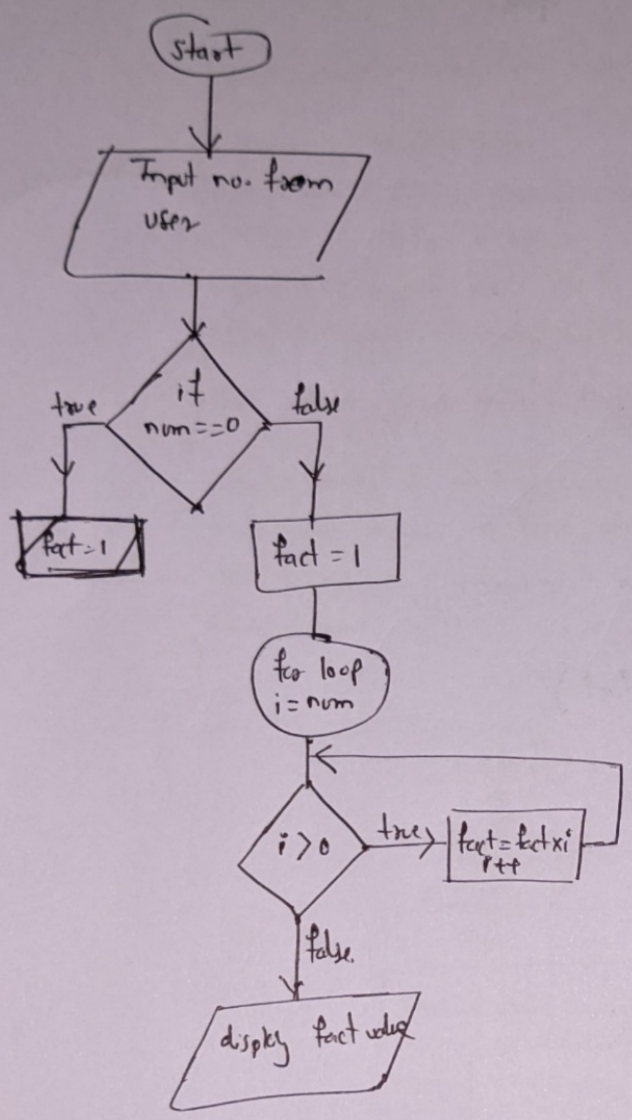
i) Check if given no. is even or odd.

Solⁿ:




```
public class EvenOdd
{
    static void isEven (int num)
    {
        if ((num % 2) == 0)
        {
            sout( num + " is even");
        }
        else
        {
            sout( num + " is odd");
        }
    }

    p.s.v. main(~ )
    {
        Sout ("Enter an integer no.");
        Scanner scanner = new Scanner(System.in);
        int num = ~~scanner.int~~N scanner.nextInt();
        scanner.close();
        isEven(num);
    }
```

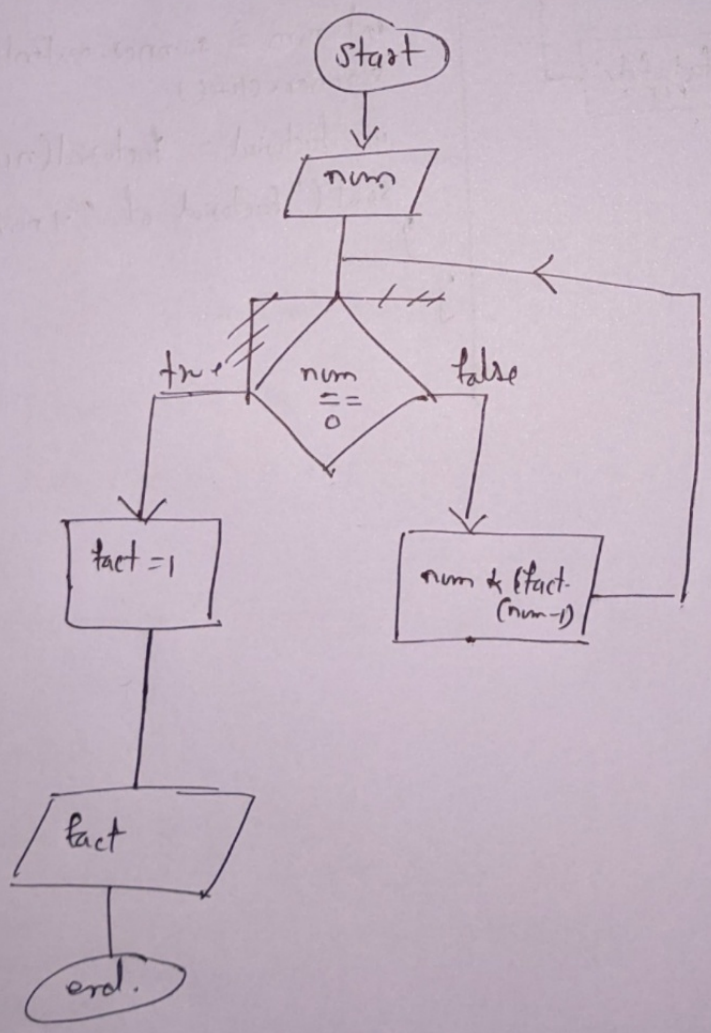
ii) Write program to find factorial of a given no.




```
public class Factorial
{
  static int factorial (int num)
  {
     int fact = 1;
     if (num == 0)
        return 1;
     for (int i = num; i>0; i--)
     {
        fact = fact * i;
     }
     return fact;
  }
  p.s.v.main (~ )
  {
   sout ("Enter an integer no.");
   Scanner scanner = new Scanner(System.in);
   int num = scanner.nextInt();
   Scanner.close();
   int factorial = factorial(num);
   sout ("Factorial of " + num + " : " + factorial)
  }
}
```

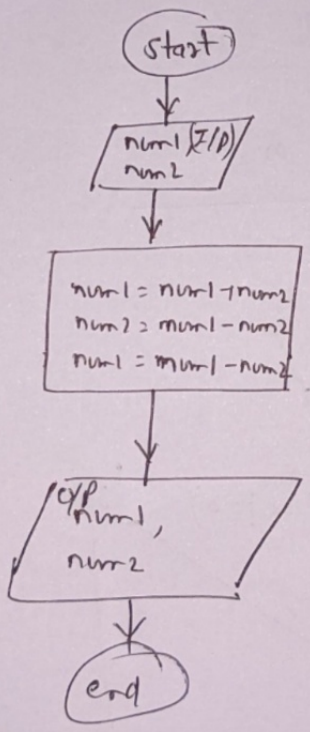
iii) find factorial using recursion

```
public class FactorialUsingRecursion
{
    static long fact(int num)
    {
        if (num==0)
            return 1;
        else
            return num * fact(num-1);
    }
    p.s.v.main(  )
    {
      sout("Enter a no.");
      int num → from user
      long fact = fact(num);
      sout("Factorial of " + num + " : " + fact);
    }
```

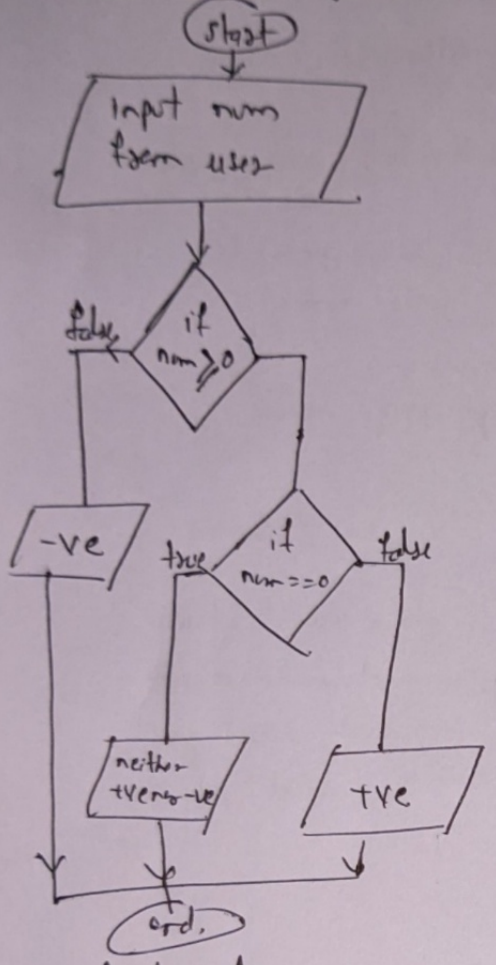


18) Swap 2 no.'s without using 3rd variable

```
public SwapNumsWithout3rdVar
{
  static void swap (int num1, int num2)
  {
    num1 = num1+num2;
    num2 = num1 - num2;
    num1 = num1 - num2;
    sout("Swapped no's are");
    sout ("num1 : "+num1 + " num2 : " + num2);
  }
  public static void main (String [] args)
  {
    sout("Enter 2 no's");
    int num1, num2 → from user
    sout ("num1 : " +num1 + " num2 : " + num2);
    swap (num1, num2);
  }
}
```

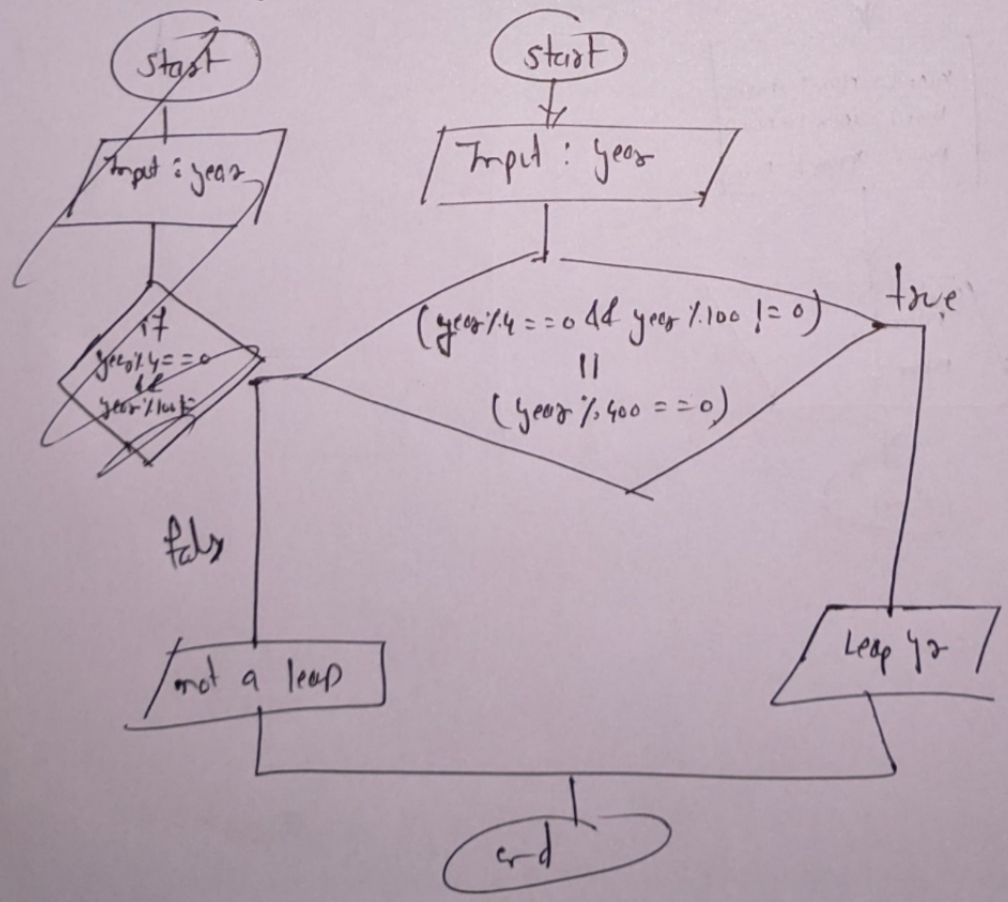


v) Check if given no. is +ve / -ve.

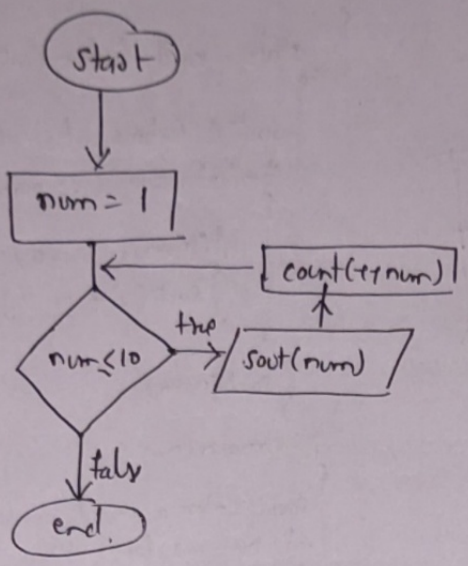



```
public class PositiveNegative
{
  static void isPosNeg(int num)
  {
    if (num == 0) sout("0 is neither +ve nor -ve");
    else if (num > 0) sout("+ve");
    else sout("-ve");
  }
  p.s.v. main(~)
  {
    int num → from user.
    isPosNeg(num);
  }
}
```

vi) find if given year is leap year or not

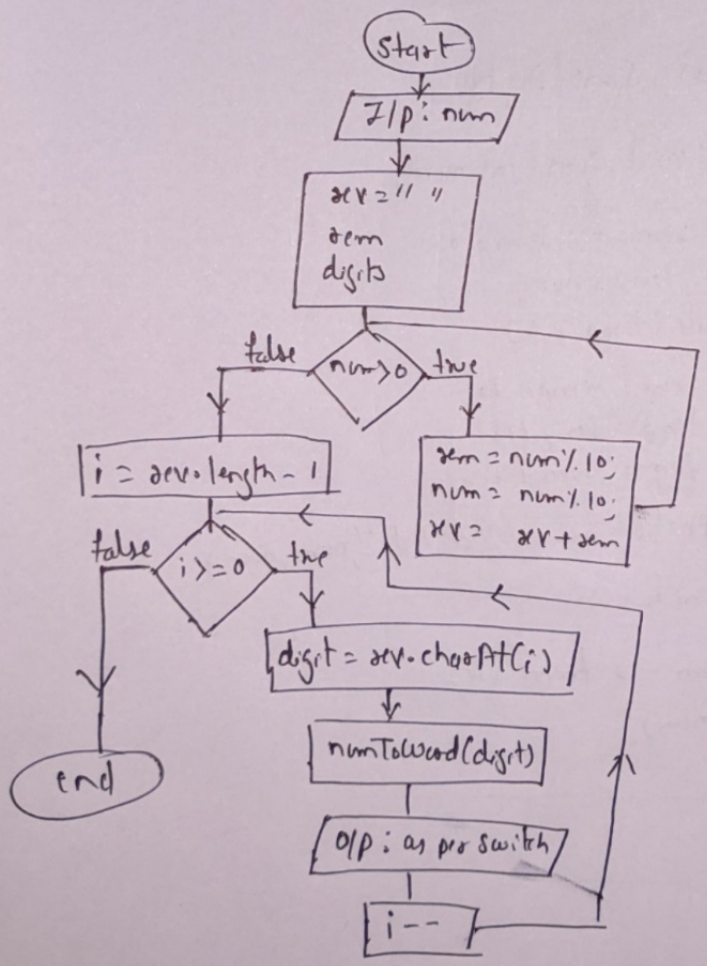


vii) Write program to print 1-10 without using loop



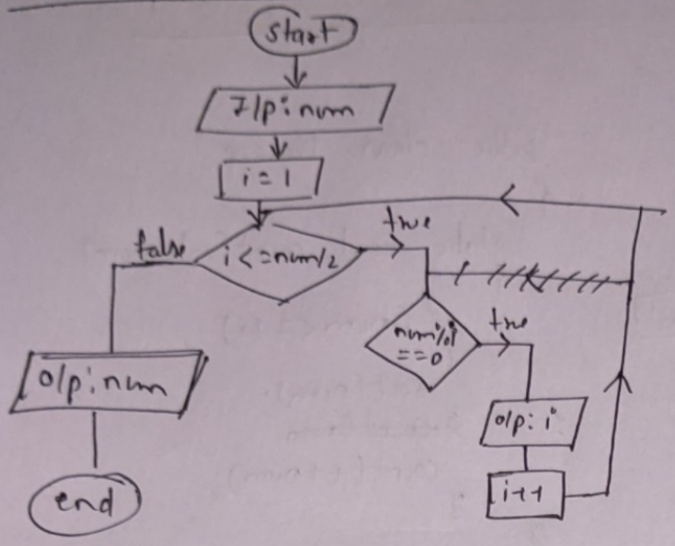

```
public class NoLoop
{
    static void count(int num)
    {
        if (num<=10)
        {
            sout(num);
            ~~count(num~~
            count(++num);
        }
    }
    p.s.v.main(~)
    {
        count(1);
    }
}
```

viii) Write program to write digits of a given no.



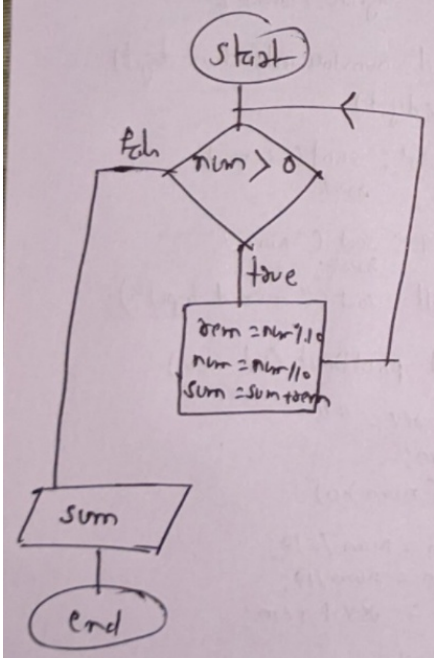

```
public class DigitsOfNumber
{
    static void numToWord(char digit)
    {
        switch(digit)
        {
            case '0': sout("zero");
                      break;
            :
            case '9': sout("nine");
                      break;
            default : sout("Invalid Input");
        }
    }
    static void printDigits(int num)
    {
        String rev = " ";
        int rem;
        while(num>0)
        {
            rem = num % 10;
            num = num / 10;
            rev = rev + rem;
        }
        char digit;
        for (int i = rev.length() - 1; i>=0; i--)
        {
            digit = rev.charAt(i);
            numToWord(digit);
        }
    }
    p.s.v.main(~)
    {
        sout("Enter a no.");
        int num --> from user
        printDigits(num);
    }
}
```

IX) write prog. to all the factors a given no.




```
public class FactorOfNumber
{
    static void factor (int num).
    {
        sout ("Factors of " +num);
        for(int i=1; i <= num/2 ; i++)
        {
            if((num%i)==0)
            {
                sout (i + " ");
            }
        }
        sout(num);
    }
    p.s.v.main(~)
    {
        sout("Enter a no.");
        int num -> from user.
        factor(num);
    }
}
```

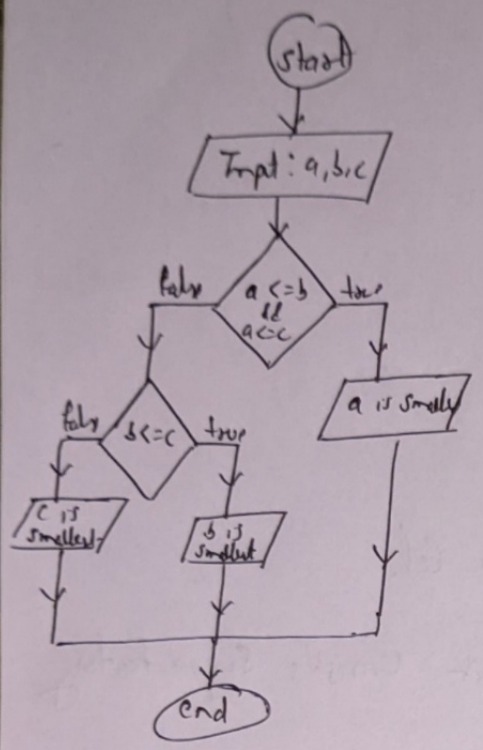
X) write a prog. to find sum of digits of given num




```
public class SumOfDigits
{
    static void sum (int num)
    {
        int rem=0 , sum = 0;
        int temp = num;
        while (temp > 0)
        {
            rem = temp% 10;
            temp = temp /10;
            sum = sum + rem;
        }
        sout.printf("Sum of %d %d :",num,sum);
    }
    p.s.v.main(~)
    {
        int num -> from user
        sum(num);
    }
}
```

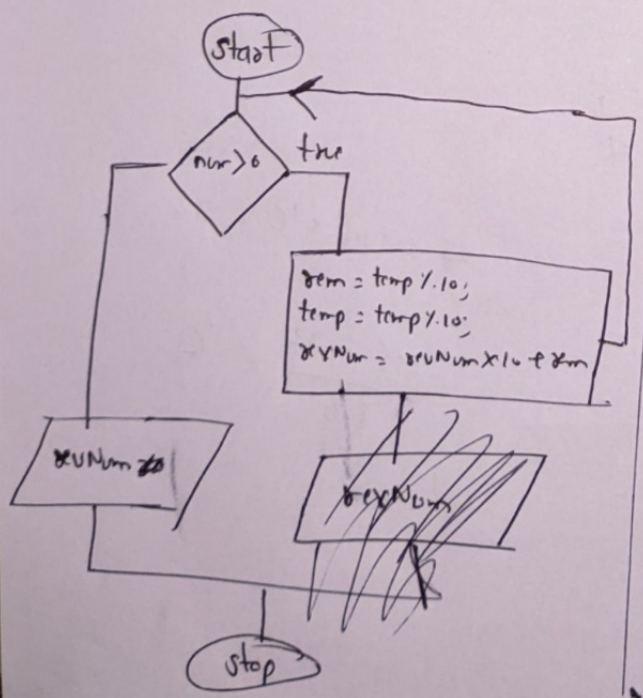
Ass 1

xi) find smallest of 3 numbers (a,b,c)




```
public class SmallestNumber
{
    static void smallestNum(int a, int b, int c)
    {
        if ( a <= b && a <= c)
        {
            sout(a + " is smallest");
        }
        else if ( b <= c)
        {
            sout(b + " is smallest");
        }
        else
        {
            sout(c + " is smallest");
        }
    }
    p.s.v.main(~)
    {
        Scanner sc = new Scanner(System.in);
        Sout("Enter 3 no.s");
        int a, b, c -> from user.
        smallestNum (a, b, c);
    }
}
```

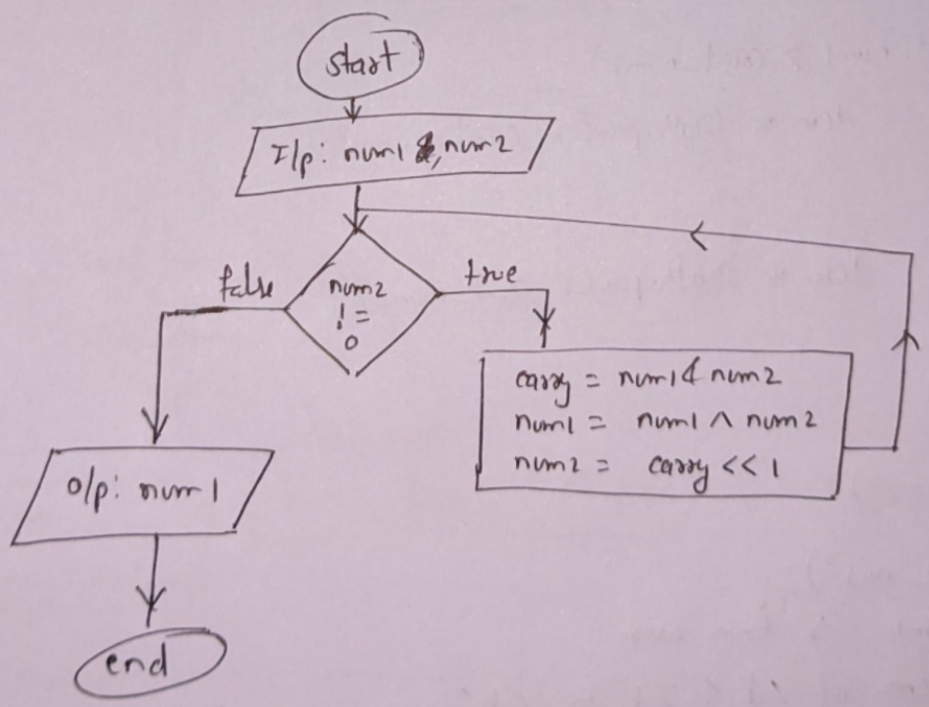
xiii) Reverse of a number



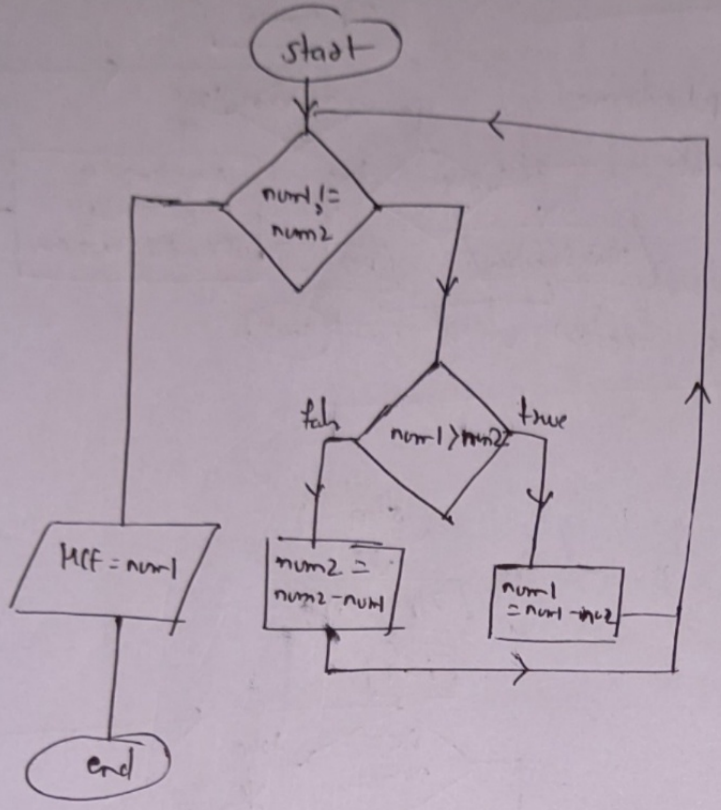

```
public class ReverseNumber
{
    static void reverseNum(int num)
    {
        int rem = 0;
        int revNum = 0;
        int temp = num;
        while (temp > 0)
        {
            rem = temp % 10;
            temp = temp / 10;
            revNum = (revNum x 10) + rem;
        }
        sout(" Reverse of " + num + " : " + revNum);
    }
    p.s.v.main(~)
    {
        int num -> from user
        sout ("Enter a num")
        reverseNum (num)
    }
}
```

xii) Add 2 no.s without arithmetic operator

```
public class AddWithoutArithmeticOperator
{
    static int add( int num1, int num2)
    {
        while( num2 != 0)
        {
            int carry = num1 & num2;
            num1 = num1 ^ num2;
            num2 = carry <<1;
        }
        return num1;
    }
    p.s.v.main(---)
    {
        Sout("Enter 2 no.s");
        int num1, num2 -> from user;
        int res = add(num1, num2);
        s.o.printf("%d + %d = %d ", num1, num2, res);
    }
}
```



xiv) HCF of 2 nos

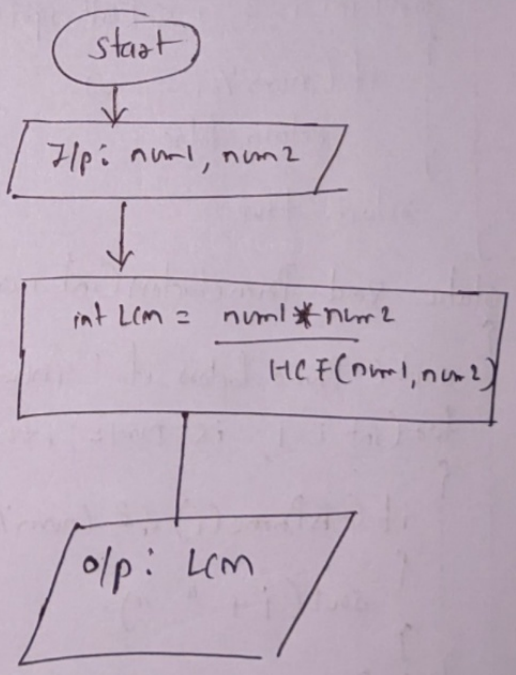



```
public class HCFofTwoNum
{
    static int HCF(int num1, int num2)
    {
        while (num1 != num2)
        {
            if (num1 > num2)
                num1 = num1 - num2;
            else
                num2 = num2 - num1
        }
        return num1;
    }
    psv main(~)
    {
        sout("Enter 2 no.'s");
        int num1, num2 → from user
        int HCF = HCF(num1, num2)
        sout("HCF of %d & %d : %d", num1, num2, HCF);
    }
}
```

xv) LCM of 2 nos

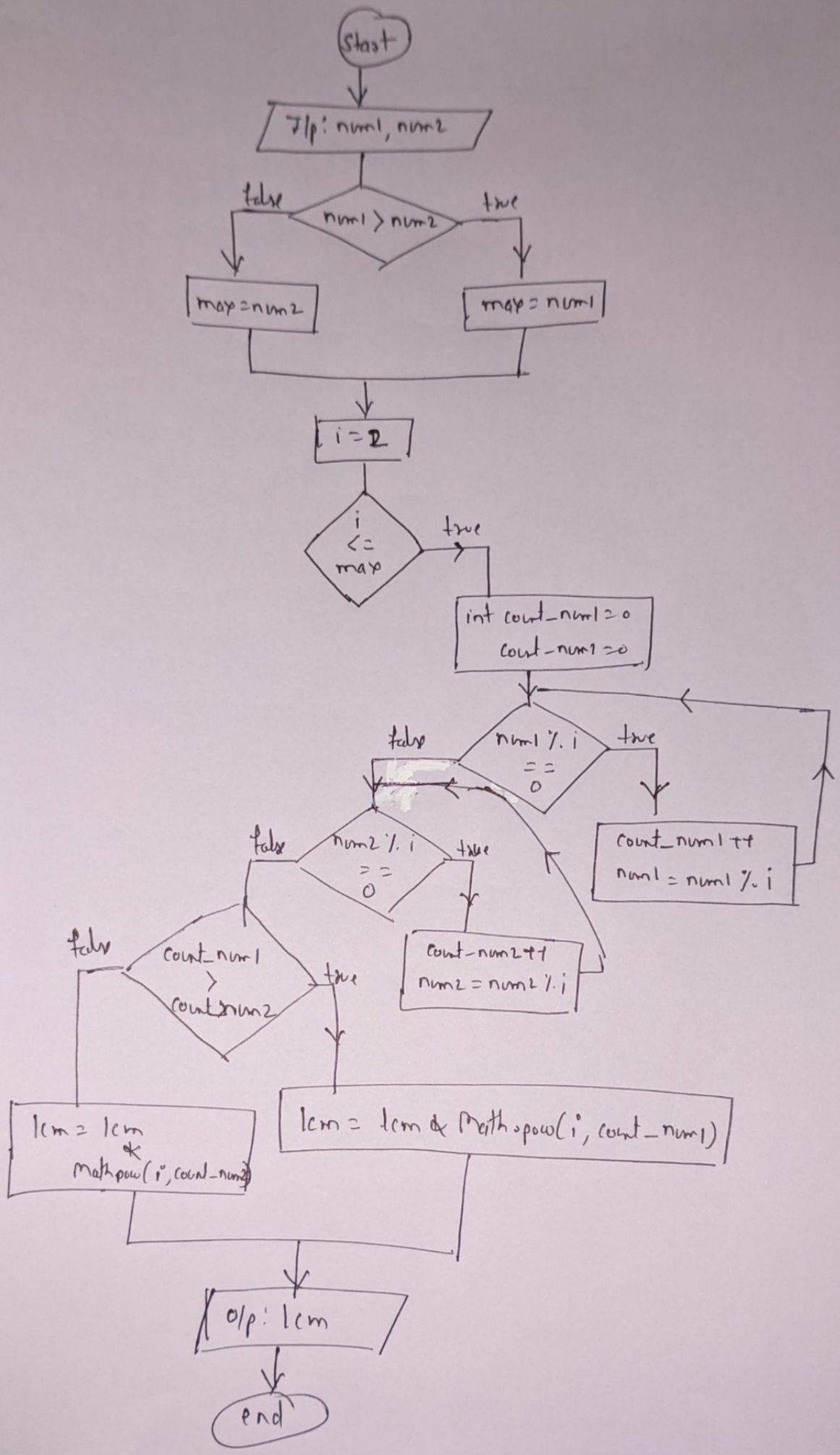
$$num1 \times num2 = HCF \times LCM$$

$$LCM = \frac{num1 \times num2}{HCF}$$



XVi> Write program to find LCM of 2 nos using Prime factor metd.

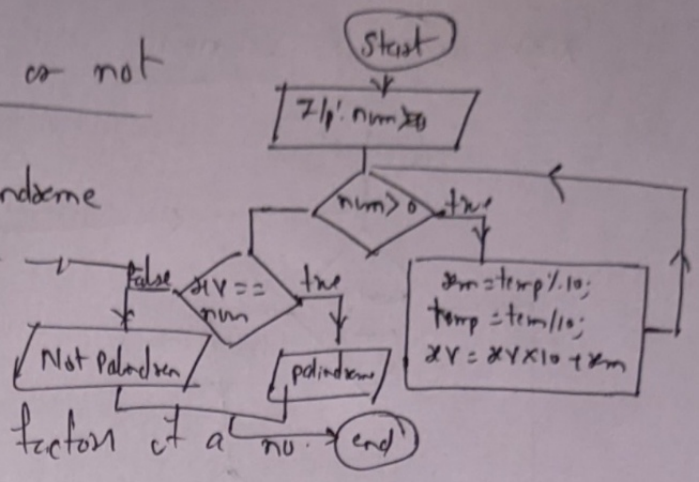
```
public class LCMUsingPrimeFactors
{
    public static int lcm (int num1, int num2)
    {
        int max = (num1 > num2) ? num1 : num2;
        int lcm = 1;

        for ( int i = 2 ; i <= max ; i++)
        {
            int count_num1 = 0 , count_num2 = 0;
            while ( num1 % i == 0)
            {
                count_num1++;
                num1 = num1/i ;
            }
            while ( num2 % i == 0)
            {
                count_num2++;
                num2 = num2/i;
            }
            if ( count_num1 > count_num2)
            {
                lcm = lcm * Math.pow(i, count_num1);
            }
            else
            {
                lcm = lcm * Math.pow(i, count_num2);
            }
        }
    return lcm;
    }

    p.s.v.main(⟶)
    {
        sod ("Enter 2 nos");
        int num1, num2 → from user
        soprint ("LCM of %d & %d = %d", num1, num2, lcm(num1, num2));
    }
}
```


Start
I/p: num1, num2
num1 > num2
false
true
max = num2
max = num1
i = 2
i <= max
true
int count_num1 = 0
count_num2 = 0
num1 % i == 0
false
true
count_num1 ++
num1 = num1 % i
num2 % i == 0
false
true
count_num2 ++
num2 = num2 % i
count_num1 > count_num2
false
true
lcm = lcm * Math.pow(i, count_num2)
lcm = lcm * Math.pow(i, count_num1)
O/p: lcm
end

xvii) Number is palindrome or not

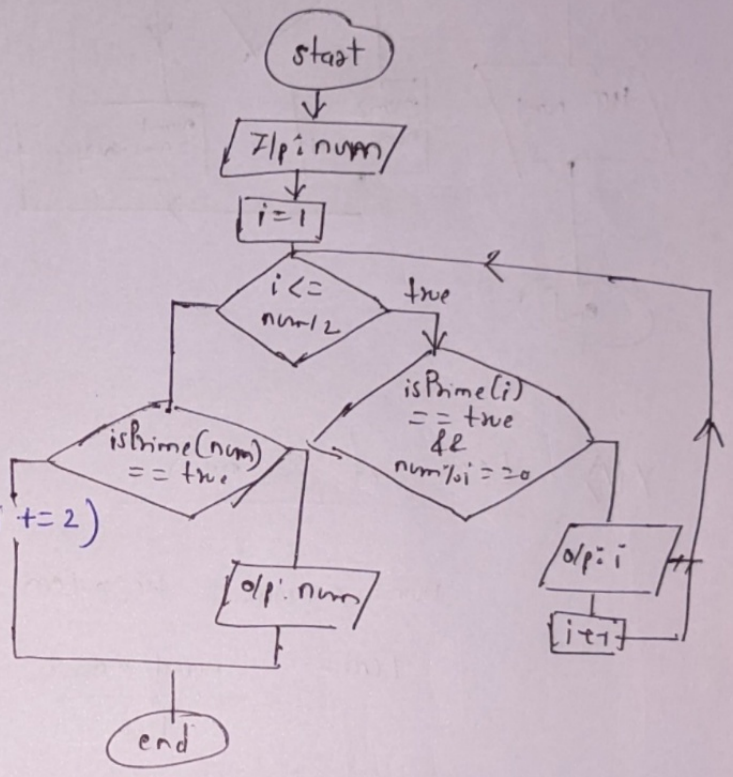
if num == xvNum ⇒ palindrome

!= ⇒ not



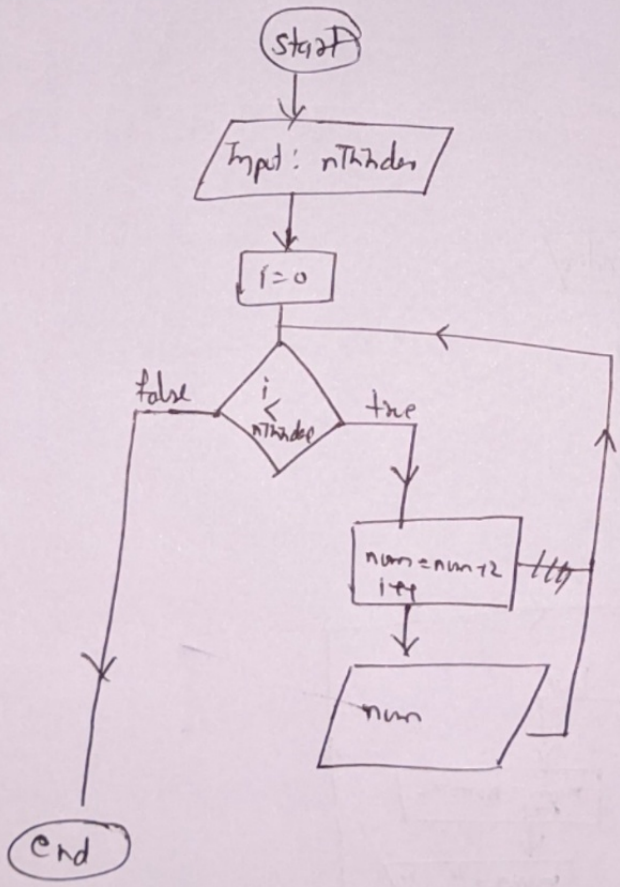
xviii) Program to write prime factors of a no.

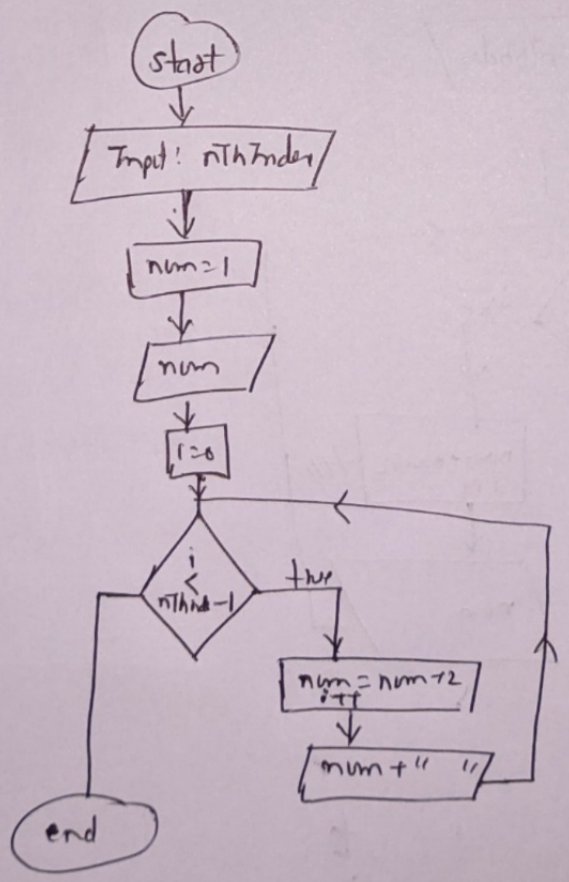
```
public class PrimeFactors
{
  static boolean isPrime (int num)
  {
    if( num <= 1)
        return false;
    else if (num == 2)
        return true;
    else if ( num % 2 == 0)
        return false;
    for (int i = 3; i < Math.sqrt(num); i += 2)
    {
      if (num % i == 0)
          return false;
    }
    return true;
  }
  static void PrimeFactors (int num)
  {
    sout ("Prime factors of " + num);
    for (int i = 1; i <= num/2; i++)
    {
      if ( isPrime(i) && (num % i) == 0)
      {
        sout( i + " ");
      }
    }
    if( isPrime(num) == true) sout(num);
  }
  p.s.v. main( — )
  {
    int num → from user
    PrimeFactors (num);
  }
}
```



xix) print EVEN Series 2, 4, 6, 8, 10, - - -

```
public class EvenSeries
{
    static void evenSeries (int nThIndex)
    {
        int num = 0;
        for (int i = 0 ; i < nThIndex ; i++)
        {
            num = num+2 ;
            soutp( num + " ")
        }
    }
    p.s.v.main(---)
    {
      sout("Enter how many terms u want of even series");
      int nThIndex -> from user
      evenSeries(nThIndex);
    }
}
```



XX) ODD Series 1, 3, 5, 7, 9, . .

```
public class OddSeries
{
    static void oddSeries( int nThIndex )
    {
        int num = 1;
        sout.print (num + " ");
        for ( int i=0 ; i<(nThIndex -1) ; i++ )
        {
            num = num+2;
            sout (num +" ");
        }
    }
    psv main(~)
    {
     sout ("Enter how many terms u want in odd series");
     int nThIndex = → from user
     oddSeries ( nThIndex );
    }
}
```