```
/* program to use linked lists for calculating factorial function
   with arbitrary precision arithmetic.  Uses java.util.LinkedList
   usage: java BigFactorial n  {n is factorial to be computed}  */

import java.util.*;
public class BigFactorial {

public static LinkedList mult(int n, LinkedList lnum) {
  LinkedList sum=new LinkedList();
  LinkedList result=new LinkedList();
  Object digit;
  int lastlnum=lnum.size() -1;
    if(lastlnum== (-1)){
        System.out.println("no elements in number!");
        System.exit(0);
    }
//  We now add lnum to itself in a loop n-1 times,
//  keeping the intermediate result each time through
    sum=lnum;  //put lnum into the sum to begin the addition loop
    for(int count=1;count<n;count++){ //do add n-1 times
      result=new LinkedList();
    int sumindex=sum.size()-1;  // # digits in sum
      int carry =0;
      for(int lindex=lastlnum;lindex>=0;lindex--){//get each digit of lnum
          digit=lnum.get(lindex);
          int x=((Integer)(digit)).intValue();
          digit=sum.get(sumindex--); //access corresponding digit in sum
          int y= ((Integer)(digit)).intValue();
          int z=x+y+carry; //add 2 digits and previous carry, call it z
          result.addFirst(new Integer(z%10)); //add (0-9) part of z
          if(z>9) carry=1; else carry=0;  //see if a carry occurs
      }
/*  when we get here, we have added lnum to sum
    now we have to do two things:
    1) copy the rest of the sum int the result as sum must be equal
       or longer than result so far
    2) propagate any carry=1 throught the remaining digits:
    e.g.  9999 + 1 = 100000 (need to modify each place possibly) */

      if(carry==1 && sumindex<0) result.addFirst(new Integer(1));
        else if(sumindex>=0 && carry==0)
            while(sumindex>=0){
                 digit=sum.get(sumindex--);
                 int y=((Integer)(digit)).intValue();
                 result.addFirst(new Integer(y));
               } else if(sumindex>=0 &&carry==1){
                   while(sumindex>=0){
                           digit=sum.get(sumindex--);
                           int y=((Integer)(digit)).intValue();
                           result.addFirst(new Integer((y+carry)%10));
                           if((y+carry)>9) carry=1; else carry=0;
                   }
                   if (carry==1) result.addFirst(new Integer(1));
              }
      sum=result; //make sum= to partial result so far, and repeat
    }
    return sum;
}
public static void main(String[] args){
  LinkedList result= new LinkedList();
  int N=Integer.parseInt(args[0]); //get argument from command line
  result.add(new Integer(1));
  for(int i=2;i<=N;i++)
    result=mult(i,result);
  System.out.println("final number is " + result);
}
}
```