// Klasinn E11 notar klasann Link sem er í Link.java.

// Lesið skilgreiningarnar og setningarnar þar.

public class E11

{

    // Notkun: splice(x,i,y);

    // Fyrir:  x er lögleg keðja með N hlekki og hlekkjarunu

    //           [x\_0,...,x\_{N-1}].

    //         0 <= i < N.

    //         y er lögleg keðja með M hlekki og hlekkjarunu

    //           [y\_0,...,y\_{M-1}].

    //         Keðjurnar x og y hafa engan sameiginlegan hlekk.

    // Eftir:  x er lögleg ke'ja með N+M hlekki og hlekkjarununa

    //           [x\_0,...,x\_i,y\_0,...,y\_{M-1},x\_{i+1},...,x\_{N-1}].

    //         Takið eftir að y rununni er splæst inn í x rununa

    //         með i+1 hlekki fyrir framan úr gömlu x rununni.

    //         Takið eftir að leyfilegt er að y sé tóm runa.

    //         Takið eftir að engir nýjir hlekkir verða til.

    public static<E> void splice( Link<E> x, int i, Link<E> y )

    {

        if(y == null) {

            return;

        }

        Link<E> rev = new Link<E>();

        Link<E> returner = new Link<E>();

        returner.head = x.head;

        Link<E> newX = x;

        int counter = 0;

        while(newX != null) {

            Link<E> temp = new Link<E>();

            temp.head = newX.head;

            temp.tail = returner;

        }

        while(newX != null){

            rev.head = newX.head;

            rev.tail = newX;

            newX = newX.tail;

        }

        x = rev;

    }

    // Notkun: Link<E> x = makeChainLoop(a);

    // Fyrir:  a er E[], ekki null.

    // Eftir:  x er lögleg keðja með N=a.length hlekki og

    //         hlekkjarunu nýrra hlekkja [h\_0,...,h\_{N-1}] þannig

    //         að h\_I.head == a[I] fyrir I=0,...,N-1.

    public static<E> Link<E> makeChainLoop( E[] a )

    {

        // Hér vantar forritstexta sem skal reikna

        // útkomuna í lykkju.

        int j = a.length-1;

        Link<E> x = new Link<E>();

        x.head = a[j];

        x.tail = null;

        j--;

        while(j >= 0)

        // x er keðja nýrra hlekkja sem innihalda a[i..a.length),

        // í þeirri röð

        // 0 <= j <= a.length

        {

            Link<E> temp = new Link<E>();

            temp.head = a[j];

            temp.tail = x;

            x = temp;

            j--;

        }

        return x;

    }

    // Notkun: Link<E> x = makeChainRecursive(a,i,j);

    // Fyrir:  a er E[], ekki null, og a[i..j) er svæði í a.

    //         (Athugið að þá er 0 <= i <= j <= a.length).

    // Eftir:  x er lögleg keðja með N=j-i hlekki og

    //         hlekkjarunu nýrra hlekkja [h\_0,...,h\_{N-1}] þannig

    //         að h\_I.head == a[I-i] fyrir I=0,...,N-1.

    public static<E> Link<E> makeChainRecursive( E[] a, int i, int j )

    {

        Link<E> x = new Link<E>();

        x.head = a[i];

        if(i == j-1) {

            return x;

        }

        x.tail = makeChainRecursive(a,i+1,j);

        return x;

    }

    // Prófið að keyra þessa skipun:

    //  java E11 1 2 3 4 5 6

    // það átti að skrifa

    //  1 2 3 4 1 2 3 4 5 6 5 6

    public static void main( String[] args )

    {

        Link<String> x = makeChainLoop(args);

        Link<String> y = makeChainRecursive(args,0,args.length);

        splice(x,3,y);

        splice(x,0,null);

        while( x != null )

        {

            System.out.print(x.head+" ");

            x = x.tail;

        }

    }

}