Alexander Guðmundsson
alg35@hi.is

```java
// Höfundur: Snorri Agnarsson, snorri@hi.is

// Listar án hliðarverkana í Java.

// Höfundur lausnar: Alexander Guðmundsson, alg35@hi.is

public class E9
{
    // Tilvik af link eru óbreytanlegir hlekkir með
    // haus sem er heiltala og hala sem er endanleg
    // keðja hlekkja.  Takið eftir að það er enginn
    // möguleiki á að breyta halanum og því eru allar
    // keðjur endanlegar.  Tóm keðja er táknuð með null.
    public static class Link
    {
        private int head;
        private Link tail;

        // Notkun: E9.Link x = new E9.Link(head,tail);
        // Fyrir:  head er heiltala, tail er E9.Link (má vera null).
        // Eftir:  x er tilvísun á nýjan E9.Link með gefinn haus og
        //         og hala.
        public Link( int head, Link tail )
        {
            this.head = head;
            this.tail = tail;
        }

        // Notkun: int h = link.head();
        // Fyrir:  link vísar á E9.Link.
        // Eftir:  h er hausinn á link.
        public int head()
        {
            return head;
        }

        // Notkun: E9.Link t = link.tail();
        // Fyrir:  link vísar á E9.Link.
        // Eftir:  t er halinn á link.
        public Link tail()
        {
            return tail;
        }
    }
```

Alexander Guðmundsson
alg35@hi.is

```
    // Notkun: E9.Link x = E9.cons(head,tail);
    // Fyrir:  head er heiltala, tail er E9.Link (má vera null).
    // Eftir:  x er tilvísun á nýjan E9.Link með gefinn haus og
    //         og hala.
    public static Link cons( int h, Link t )
    {
        return new Link(h,t);
    }

    // Notkun: int h = head(x);
    // Fyrir:  x er tilvísun á E9.Link, má ekki vera null.
    // Eftir:  h er hausinn á x.
    public static int head( Link x )
    {
        return x.head();
    }

    // Notkun: E9.Link t = tail(x);
    // Fyrir:  x er tilvísun á E9.Link, má ekki vera null.
    // Eftir:  h er halinn á x.
    public static Link tail( Link x )
    {
        return x.tail();
    }

    // Notkun: int n = E9.length(x);
    // Fyrir:  x er E9.Link tilvísun, má vera null.
    // Eftir:  n er fjöldi hlekkja í keðju x.
    public static int length( E9.Link x )
    {
        if(x == null)
        {
            return 0;
        }
        int head = x.head;
        int n = 1;
        Link tail = x.tail;

        //  Skref:  head verður haus af tail halanum
        //          tail halinn verður hali tail halans
        //          n stækkar um 1
        //  Eftir:  n er jafnt dýptinni á x.
        while(tail != null) {
            head = tail.head;
            tail = tail.tail;
```

```
            n += 1;
        }
        return n;
    }

    // Notkun: int i = E9.nth(x,n);
    // Fyrir:  x er keðja með a.m.k. n+1 hlekki.
    // Eftir:  i er hausinn á n-ta hlekk í keðjunni
    //         þar sem 0-ti hlekkur er fremsti hlekkur.
    public static int nth( E9.Link x, int n )
    {
        int i = 0;

        //  Skref:  x er jafn halanum af x
        //          i hækkar um 1
        //  Eftir:  i er jafnt n
        //          hausinn á x er gildi númer n á
        //          upprunarlega x
        while(i != n) {
            x = x.tail;
            i += 1;
        }
        return x.head;
    }

    // Notkun: E9.Link x = makeChain(a);
    // Fyrir:  a er tilvísun á int[]. Má ekki vera null
    //         en má vera tómt.
    // Eftir:  x er keðja sem inniheldur gildin í a
    //         þannig að fyrir i=0,...,a.length gildir
    //         E9.nth(x,i) == a[i].
    public static Link makeChain( int[] a )
    {
        int head = a[a.length-1];
        Link x = new Link(head,null);

        //  Skref:  hali x verður að x
        //          haus x verður að a[i]
        //  Eftir:  x verður að Link af a.
        for(int i = a.length-2; i >= 0; i--) {
            x = cons(a[i],x);
        }
        return x;
```

```
}

// Notkun: int i = E9.last(x);
// Fyrir:  x er tilvísun á E9.Link, má ekki vera null.
// Eftir:  i er gildið í (hausinn á) aftasta hlekk x.

public static int last( Link x )
{
    //  Skref:  x verður að hala x
    //  Eftir:  aðeins haus eftir í x sem er
    //          síðasta gildið
    while(x.tail != null) {
        x = x.tail;
    }
    return x.head;
}

// Notkun: E9.Link z = E9.removeLast(x);
// Fyrir:  x er tilvísun á E9.Link, má ekki vera null.
// Eftir:  z er keðja sem inniheldur nýja hlekki
//         þannig að E9.length(z) == E9.length(x)-1
//         og fyrir i=0,...,E9.length(z) gildir
//         E9.nth(z,i) == E9.nth(x,i).
public static Link removeLast( Link x )
{
    int length = length(x);
    int nth = nth(x,length-1);
    Link z = new Link(nth, null);
    length = length - 1;

    //  Skref:  nth verður gildi númer length-1
    //          hali af z verður að z
    //          haus af z verður að nth
    //          length minkar um 1
    //  Eftir:  lengd er orðin 0
    //          lengd á z er lengd á x - 1
    //          z er x án seinasta hala
    while(length > 0) {
        nth = nth(x,length-1);
        z = cons(nth,z);
        length = length - 1;
    }
    return z;
```

```
}

// Notkun: E9.Link r = E9.reverse(x);
// Fyrir:  x er keðja, má vera tóm.
// Eftir:  z er jafn löng keðja og x, þannig að
//         fyrir i=0,...,E9.length(x)-1 gildir
//         E9.nth(x,i) == E9.nth(r,E9.length(x)-i-1).
public static Link reverse( Link x )
{
    int head = x.head;
    Link xTemp = x.tail;
    Link z = new Link(head,null);

    //  Skref:  haus head verður hausinn á xTemp
    //          haus z verður haus head
    //          hali z verður z
    //          xTemp keðjan verður halinn á xTemp
    //  Eftir:  z er jafn djúpt og x
    //          nth(x,i) == nth(z,length(x)-i-1)
    while(xTemp != null) {
        head = xTemp.head;
        z = cons(head,z);
        xTemp = xTemp.tail;
    }
    return z;
}

// Keyrið skipunina
//   java E9 1 2 3 4
// og sýnið hvað forritið skrifar
public static void main( String[] args )
{
    E9.Link x = null;
    for( int i=0 ; i!=args.length ; i++ )
        x = E9.cons(Integer.parseInt(args[i]),x);
    while( x != null )
    {
        E9.Link z = reverse(x);
        x = z;
        while( z != null )
        {
            System.out.print(z.head); System.out.print(" ");
            z = z.tail;
        }
```

Alexander Guðmundsson
alg35@hi.is

```
            x = removeLast(x);
            System.out.println();
        }
    }
}
```

```
PS C:\Users\alexa\OneDrive\Desktop\Háskóli Íslands 2\Master\1 ár\vor\Rökstudd forritun\Heimaverkefni 9> java E9 1 2 3 4
1 2 3 4
4 3 2 1
1 2 3 4
4 3 2 1
1 2 3 4
4 3 2 1
```

-
-
-