

```
public class Test
```

```
{ int data;
```

```
public Test(int x ) { data=x; }
```

```
public void assign(Test b, Test c, int d)
```

```
{ data= 2;
```

```
  b= new Test(2);
```

```
  c.data= 2;
```

```
  d=2;      }
```

What does
this print?

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

```
{ Test a, b, c; int d;
```

```
  a= new Test(1); b= new Test(1);
```

```
  c= new Test(1); d=1;
```

```
  a.assign(b, c, d);
```

```
  System.out.println(a.data + " " + b.data + " "  
                      + c.data + " " + d); }
```

```

public class Test
{
    int data;

    public Test(int x )
    {
        data=x;
    }

    public void assign(Test b, Test c, int d)
    {
        data= 2;
        b= new Test(2);
        c.data= 2;
        d=2;
    }
}

```

```

public static void main(String [] args)
{
    Test a, b, c;
    int d;

    a= new Test(1);
    b= new Test(1);
    c= new Test(1);
    d=1;

    a.assign(b, c, d);

    System.out.println(a.data + " " +
                        b.data + " " +
                        c.data + " " +
                        d);
}
}

```

On slides: I may use awkward spacing in order to allow visibility/increased font size. Please format your code nicely! Also, include lots of comments!

StartupSlam
Sept 30, 2017
startupslam.io



Register before Sept 9th to secure a free t-shirt!



Office Hours

Come spend some 1:1 time with Microsoft engineers! Bring your resume for review, prep for your upcoming interview, get more information on what you need to know to be an awesome Microsoft candidate or just drop by to say hi!

Time

Wednesday, September 20th, 1-4pm

Location

Engineering/Computer Science, Room 227

Microsoft Tech Talk

Come meet some amazing Microsoft engineers and alumni! We will be discussing life at Microsoft, exciting new technologies, as well as career and internship opportunities. Come to ask questions, grab dinner, and bring your resume!

Time

Wednesday, September 20th, 5:00pm

Location

Engineering/Computer Science, Room 660



MSFT3c – Coding Competition

Bring your laptop, charger and up to two friends for your chance to win a gift card or other great prizes.

Refreshments will be provided.

No purchase necessary, open to students 18+

For Official Rules and a full list of events, see MSFT3c.com

Time

Thursday, September 21st, 6-9pm

Location

Engineering/Computer Science, Room 660

We will be recruiting for Full-time Software Engineering and Program Manager roles, as well as for internships for both CS students, and students interested in the field of CS, through our Microsoft Explore Program.

Announcements

Assignment #1:

Part 1A Programming Questions: Upload to connex by before Saturday Sept. 23 at 11:55pm. You are allowed to resubmit an unlimited number of times up until the deadline.

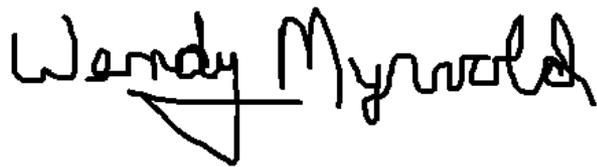
Part 1B Written Questions: Hand in on paper at the beginning of class on Thursday September 28.

Make sure you sign the attendance sheet today to get credit for attending.

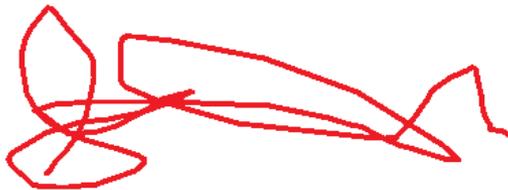
Powerpoint slides will be posted: click on the "Selected class notes" link on the course web page.

Students are expected to attend all the classes. In order to write the final exam, **students should not miss more than 6 classes** with the exception of extreme circumstances with appropriate documentation (for example, a note from a health care provider).

Your signature on the attendance sheet must be legible:



OK



NOT OK

If you want credit for attendance I should be able to read your name from what you have written.

DO NOT SIGN the sheet for a friend.

Lab #1 is posted: Your first lab will be Sept. 11, 12, or 15.

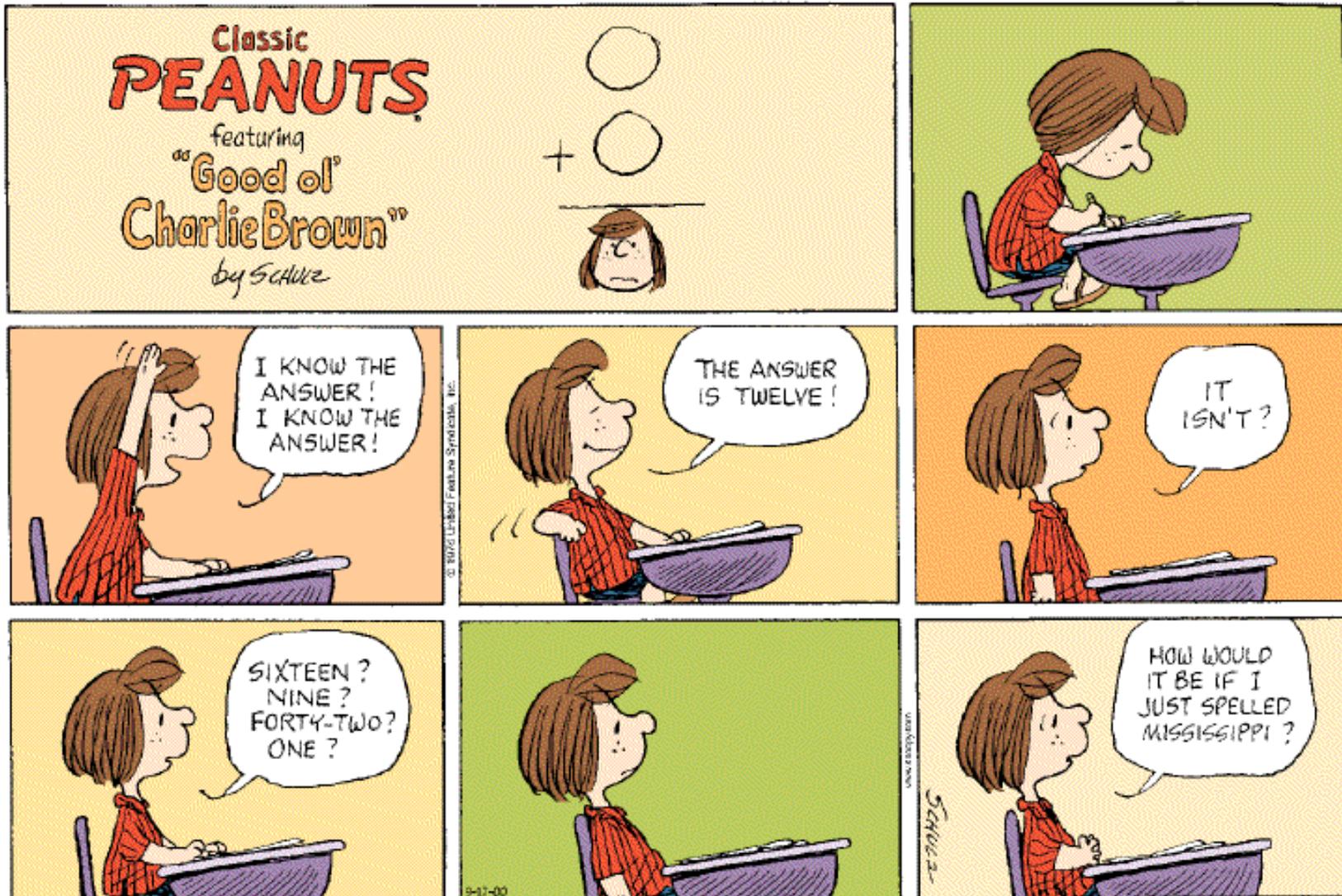
Try to complete the exercises posted before attending the lab. Attend the lab even if you do not find time to complete everything beforehand.

Lab sections:

B01	ECS 258	M	13:30-14:20
B02	ECS 258	M	14:30-15:20
B03	ECS 258	M	15:30-16:20
B04	ECS 258	T	09:30-10:20
B05	ECS 258	T	10:30-11:20
B06	ECS 258	F	13:30-14:20
B07	ECS 258	F	14:30-15:20

Labs will help you do your assignments and exams but there is nothing to hand in and they are not marked.

You will learn a lot more if you try the problems and get them wrong than if you do not try. Also, it helps me know where the class is in terms of understanding.



Their impact is broad and far-reaching.

Internet. Web search, packet routing, distributed file sharing, ...

Biology. Human genome project, protein folding, ...

Computers. Circuit layout, file system, compilers, ...

Computer graphics. Movies, video games, virtual reality, ...

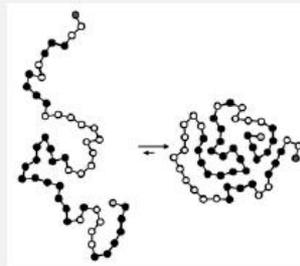
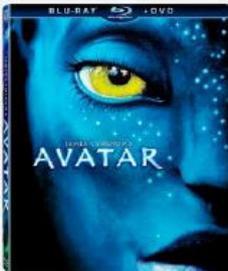
Security. Cell phones, e-commerce, voting machines, ...

Multimedia. MP3, JPG, DivX, HDTV, face recognition, ...

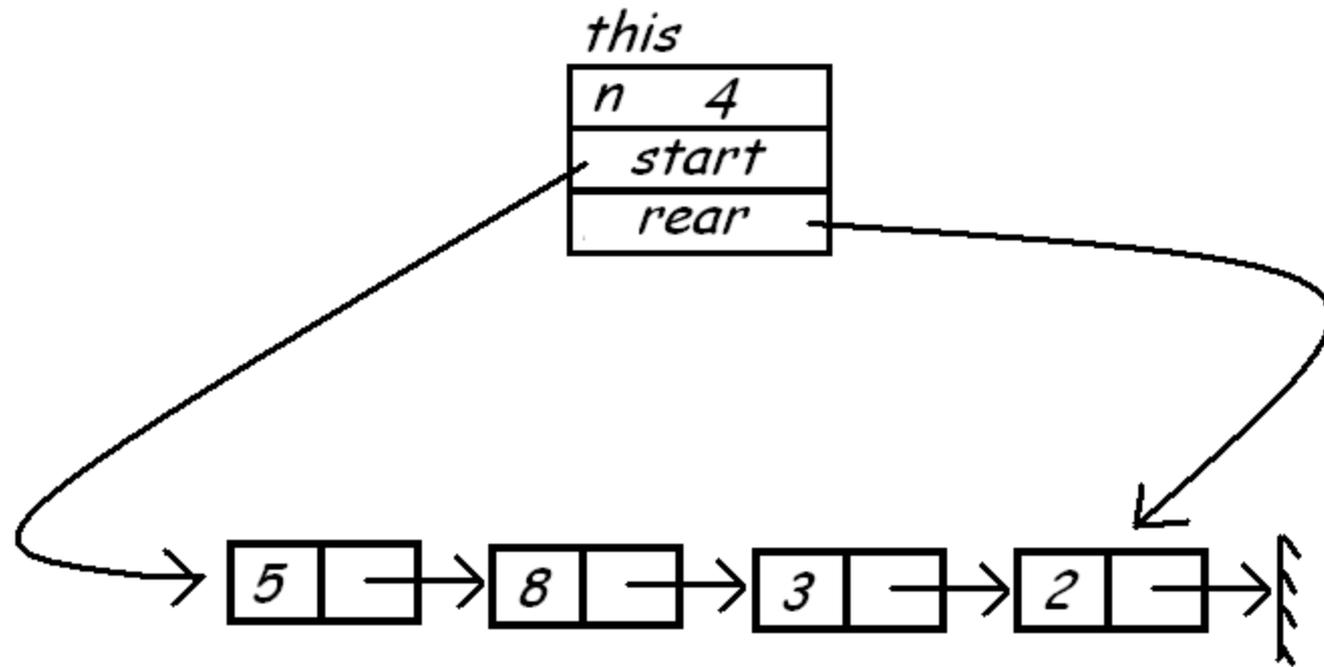
Social networks. Recommendations, news feeds, advertisements, ...

Physics. N-body simulation, particle collision simulation, ...

...



Review of linked lists



How can we split a list into two lists for a Mergesort?

One solution: `evenOddSplit()`

Learning objectives: Understanding of how to program linked lists in Java. To learn how to draw pictures of what a program is doing as it is executing (this will help you to more easily write correct code and to debug code which is not correct).

Testing evenOddSplit:

Walk through some test examples by hand to make sure you know how to step through code and draw pictures of what is happening as it is executing.

Programs most often have mistakes with extremal cases. What happens with lists with 0, 1, and 2 items?

The code is doing different things with even numbered and odd numbered cells of the list. Does it work properly for lists of even length? Does it work properly for lists of odd length?

Fix any bugs you find.

Using the data structures for the lab:

```
class ListNode {
```

```
    public int data;
```

```
    public ListNode next;
```

```
    public ListNode(int x, ListNode ptr)
```

```
{
```

```
        data= x;
```

```
        next= ptr;
```

```
}
```

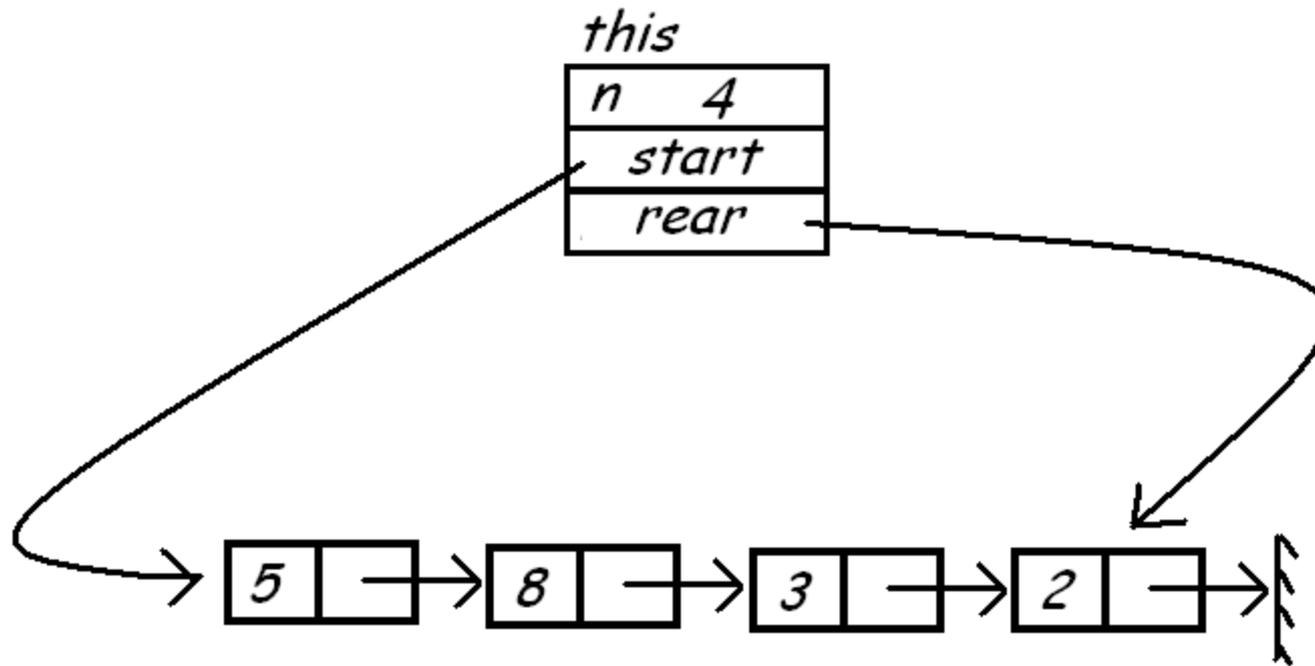
```
}
```

```
public class LinkedList
{
    int n;
    ListNode start;
    ListNode rear;

    public LinkedList()
    {
        n= 0;
        start= null;
        rear= null;
    }
}
```

```
public class SplitList
{
    LinkedList list1;
    LinkedList list2;

    public SplitList()
    {
        list1= new LinkedList();
        list2= new LinkedList();
    }
}
```



Step through the code in `evenOddSplit()` with this example.

```
public SplitList evenOddSplit()  
{ SplitList result;
```

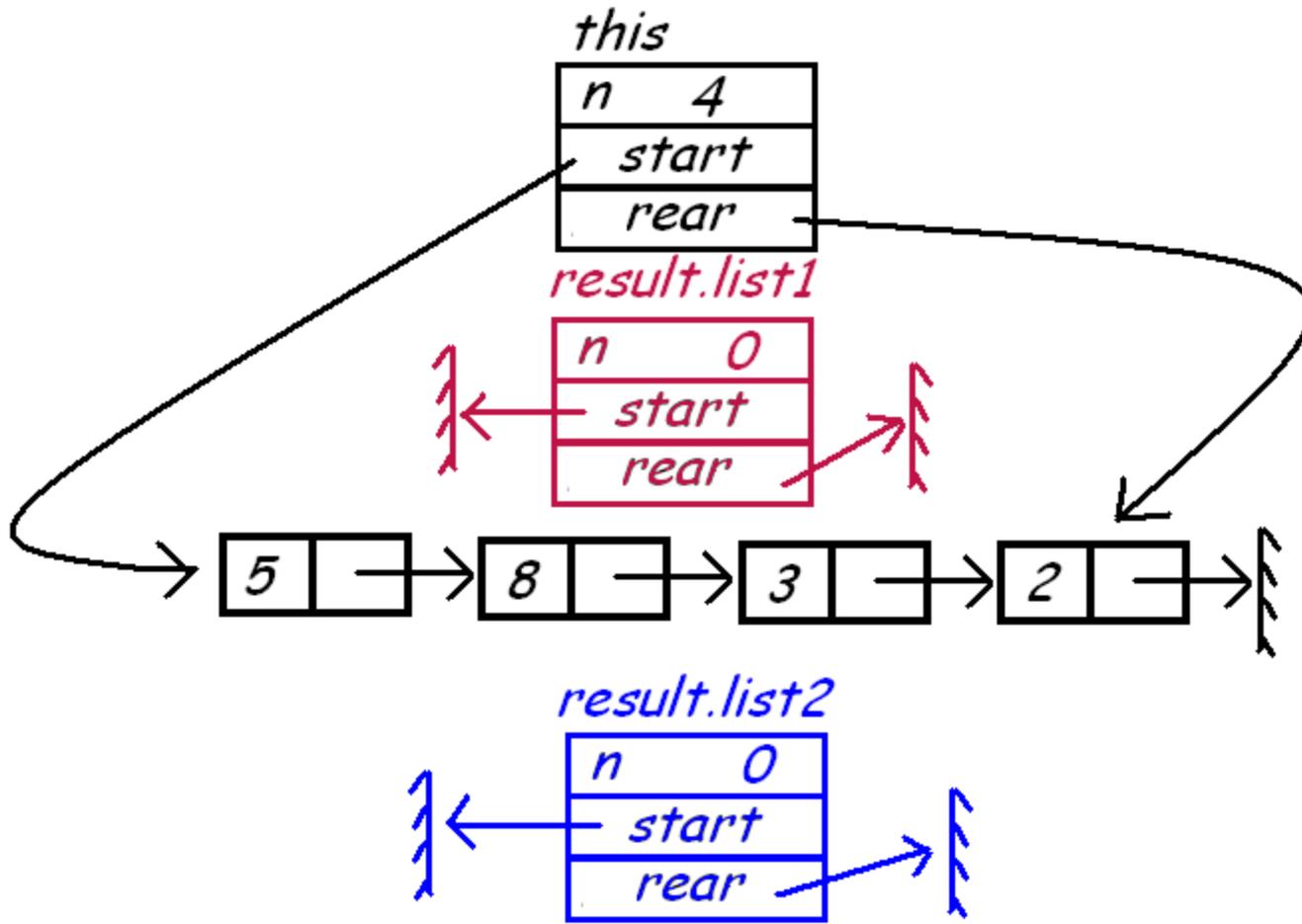
```
    result = new SplitList();  
    if (start == null) return(result);  
    result.list1.start = start;  
    result.list1.rear = start;  
    result.list1.n++;
```

```
    if (start.next == null) return(result);  
    result.list2.start = start.next;  
    result.list2.rear = start.next;  
    result.list2.n++;
```

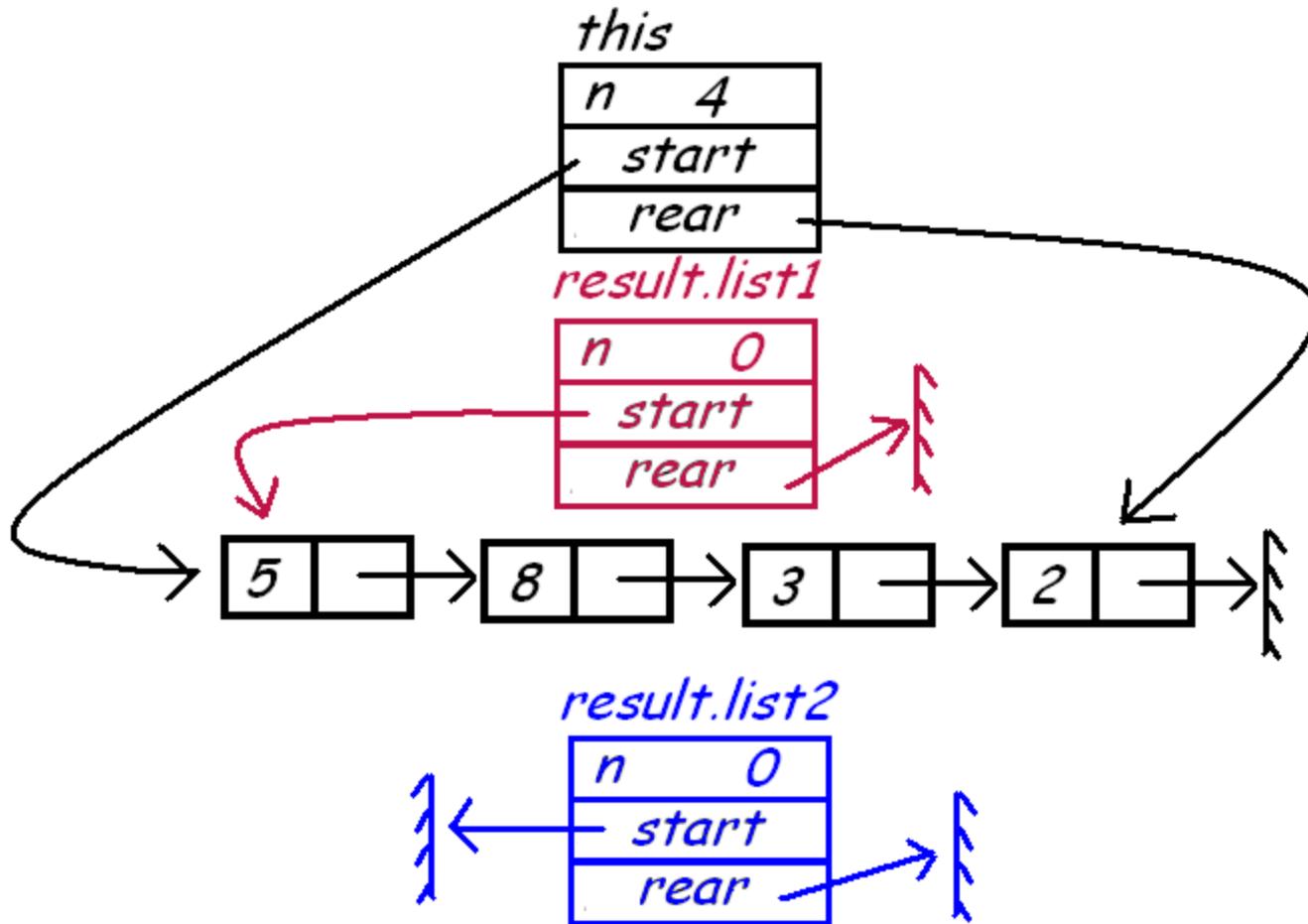
Buggy
code

```
while (result.list2.rear.next!= null )
{
    result.list1.rear.next= result.list2.rear.next;
    result.list1.rear= result.list2.rear.next;
    result.list1.n++;
    if (result.list1.rear.next != null)
    {
        result.list2.rear.next= result.list1.rear.next;
        result.list2.rear= result.list1.rear.next;
        result.list2.n++;
    }
}

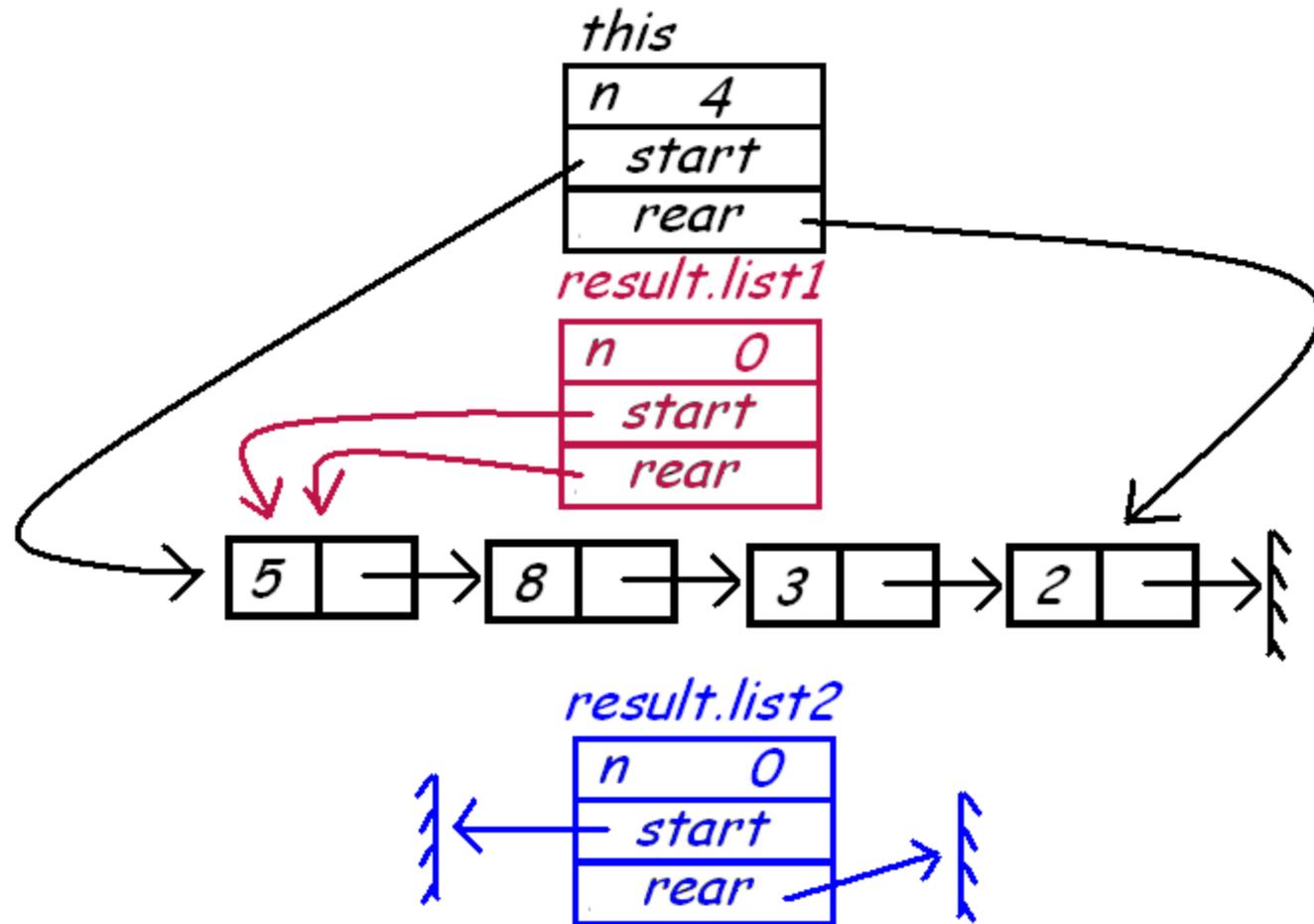
result.list1.rear.next= null;
result.list2.rear.next= null;
return(result);
```



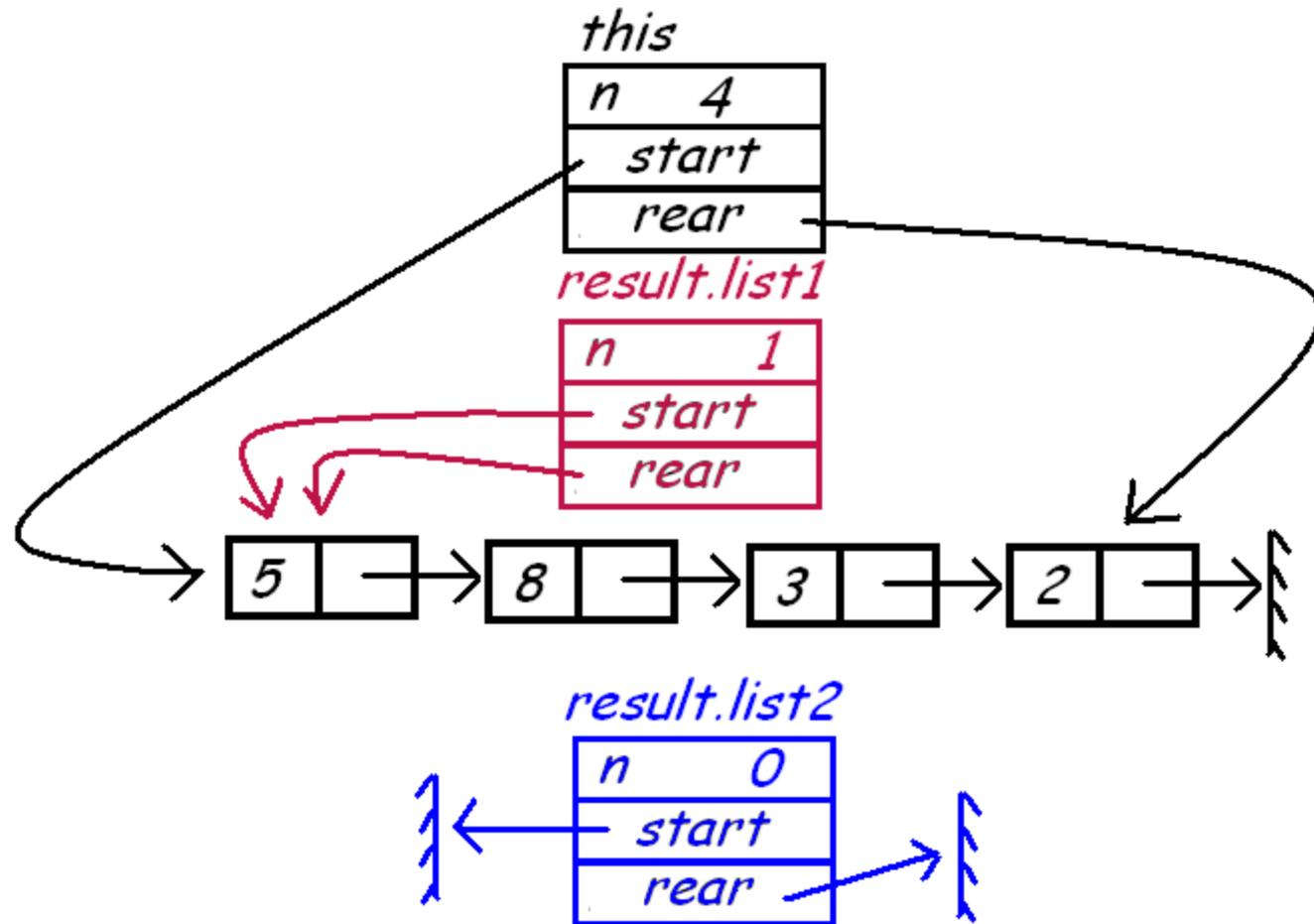
result= new SplitList();



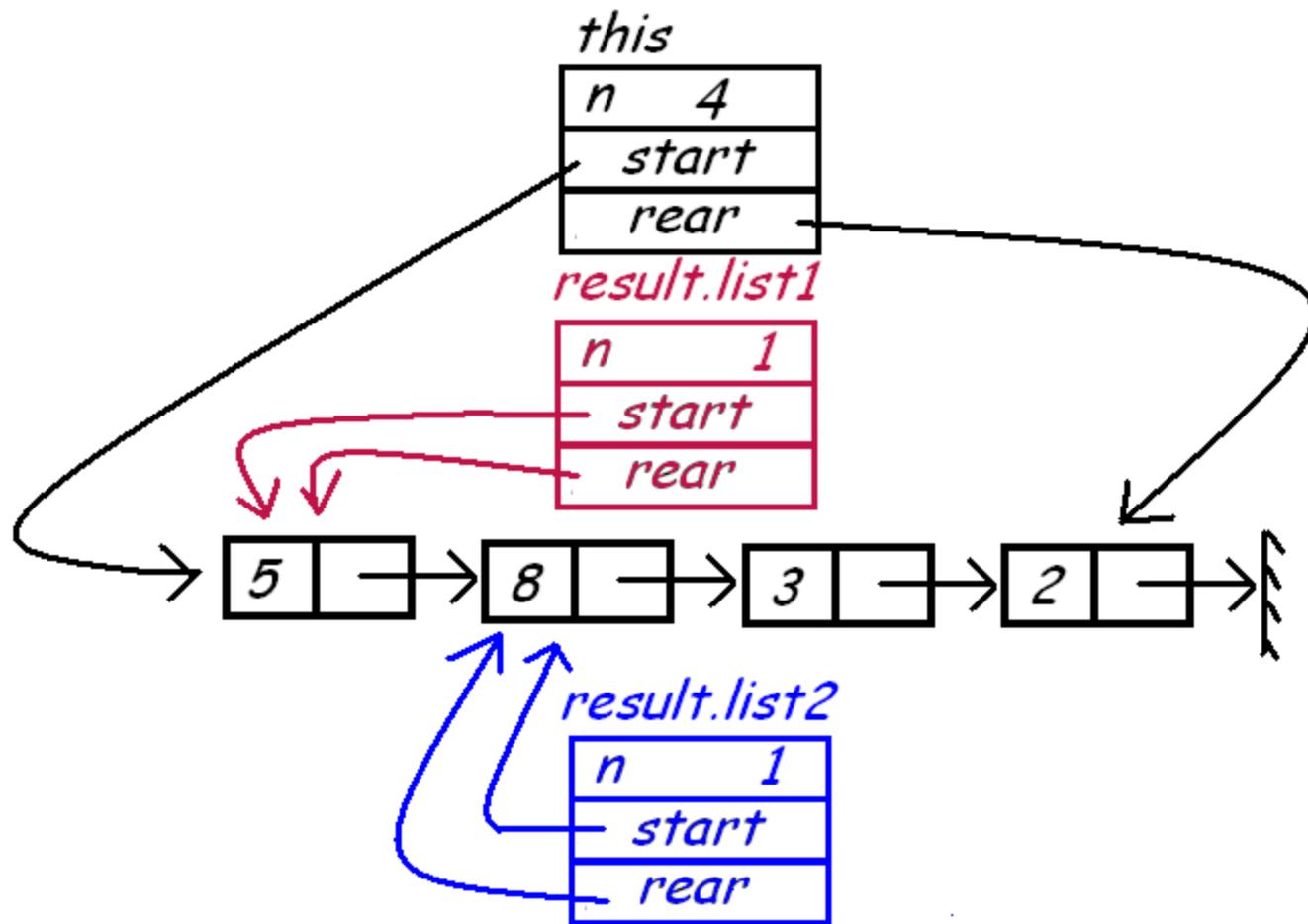
`result.list1.start= start;`



`result.list1.rear = start;`



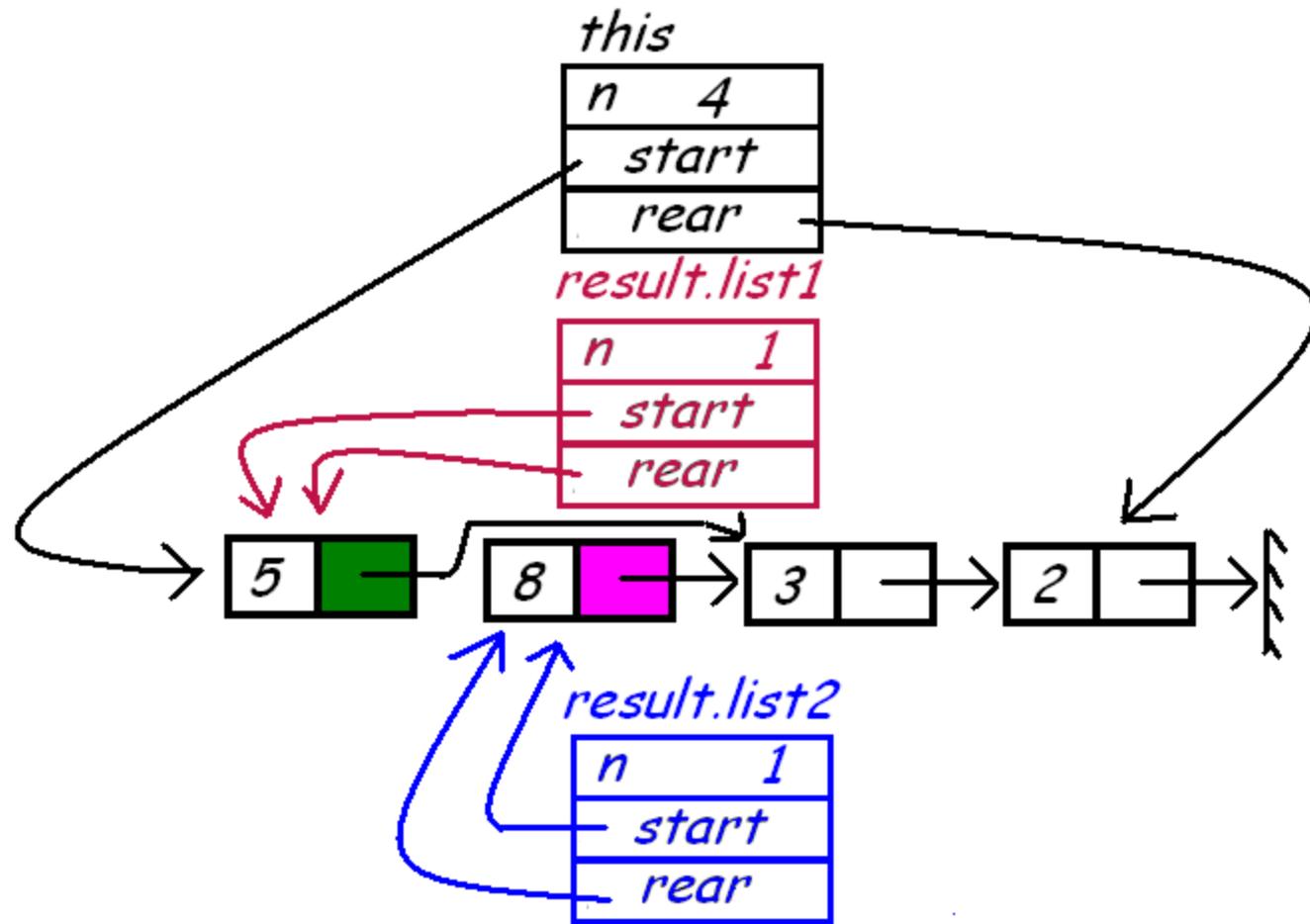
`result.list1.n++;`



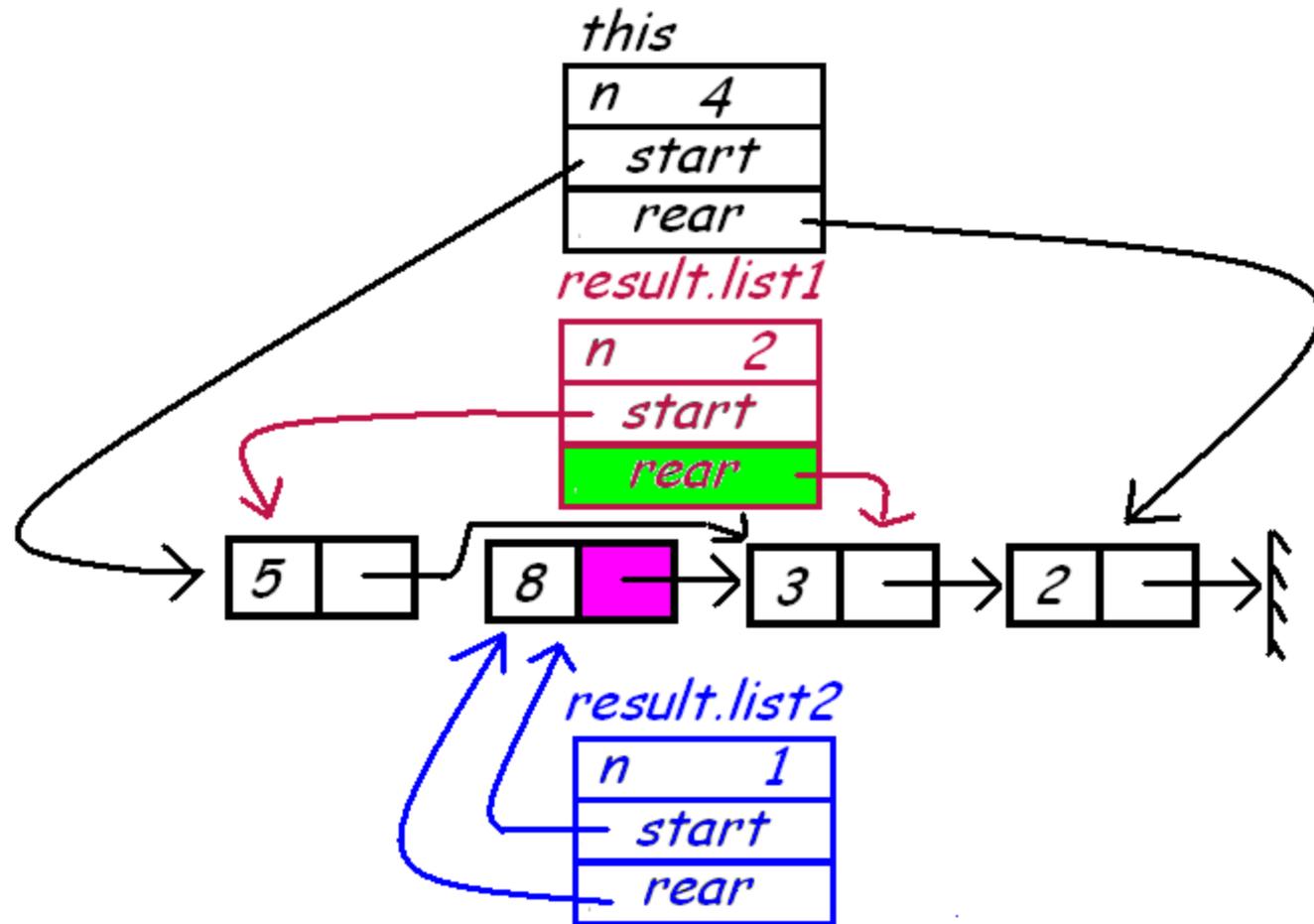
```

result.list2.start= start.next;
result.list2.rear = start.next;
result.list2.n++;

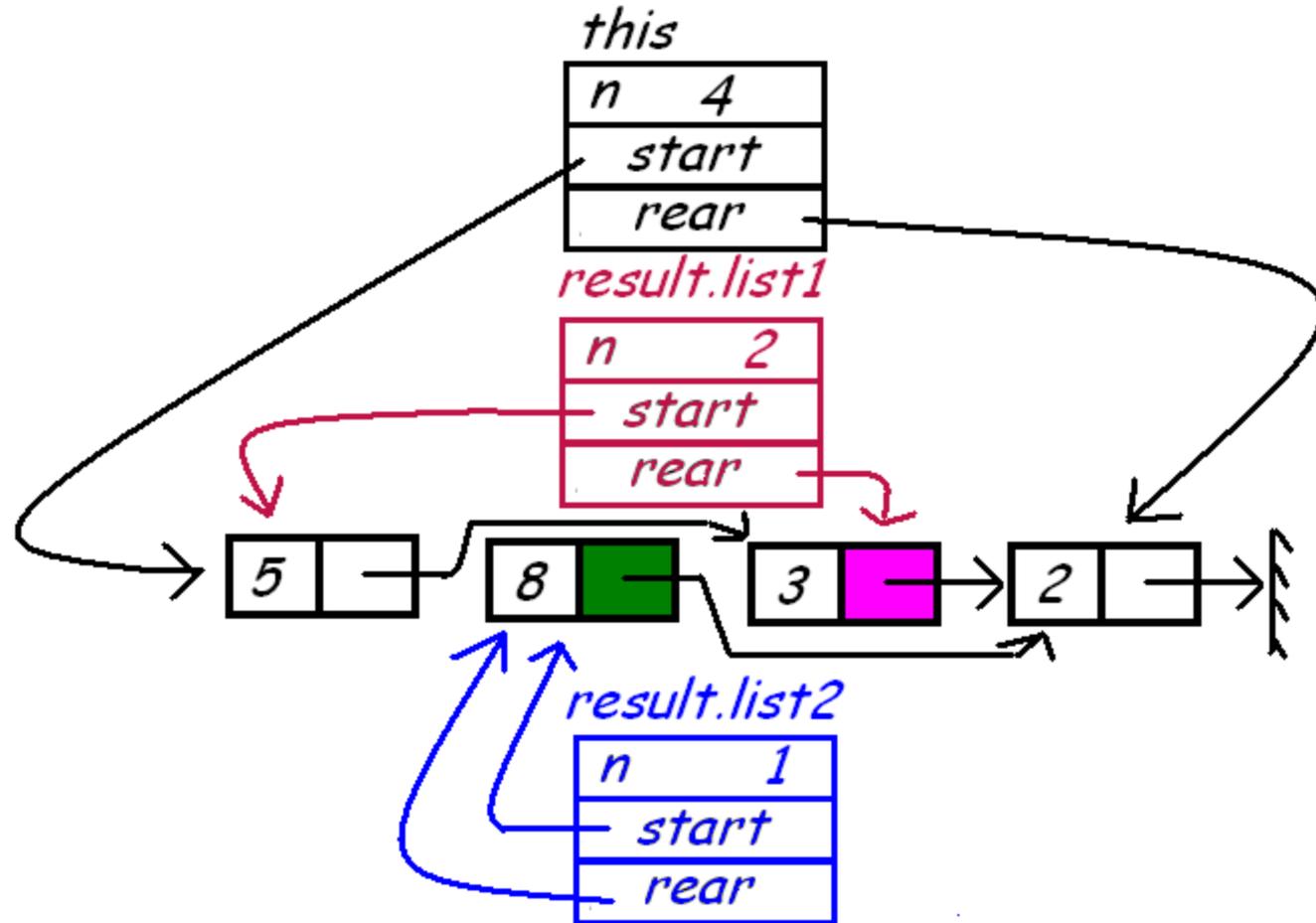
```



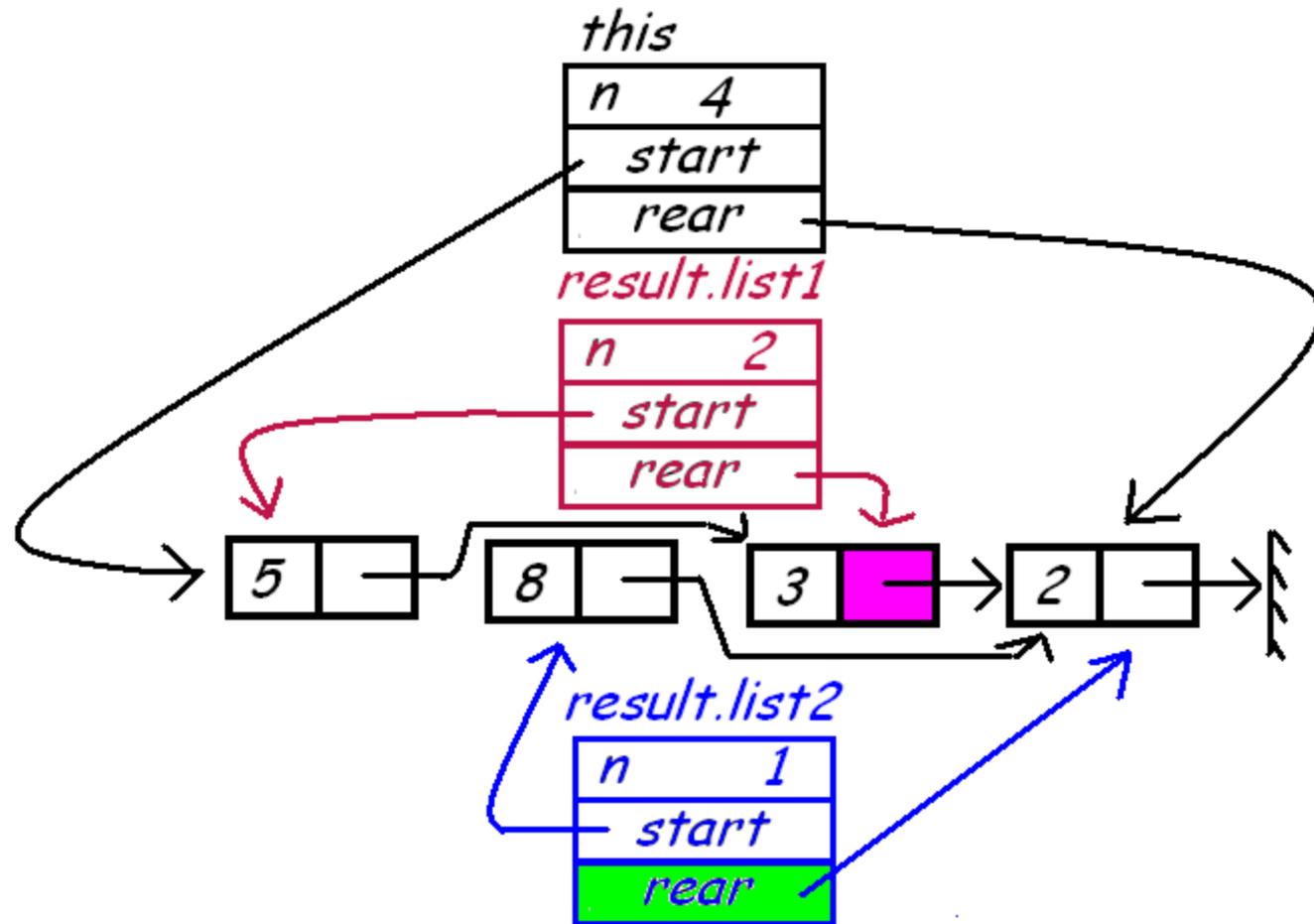
`result.list1.rear.next = result.list2.rear.next;`



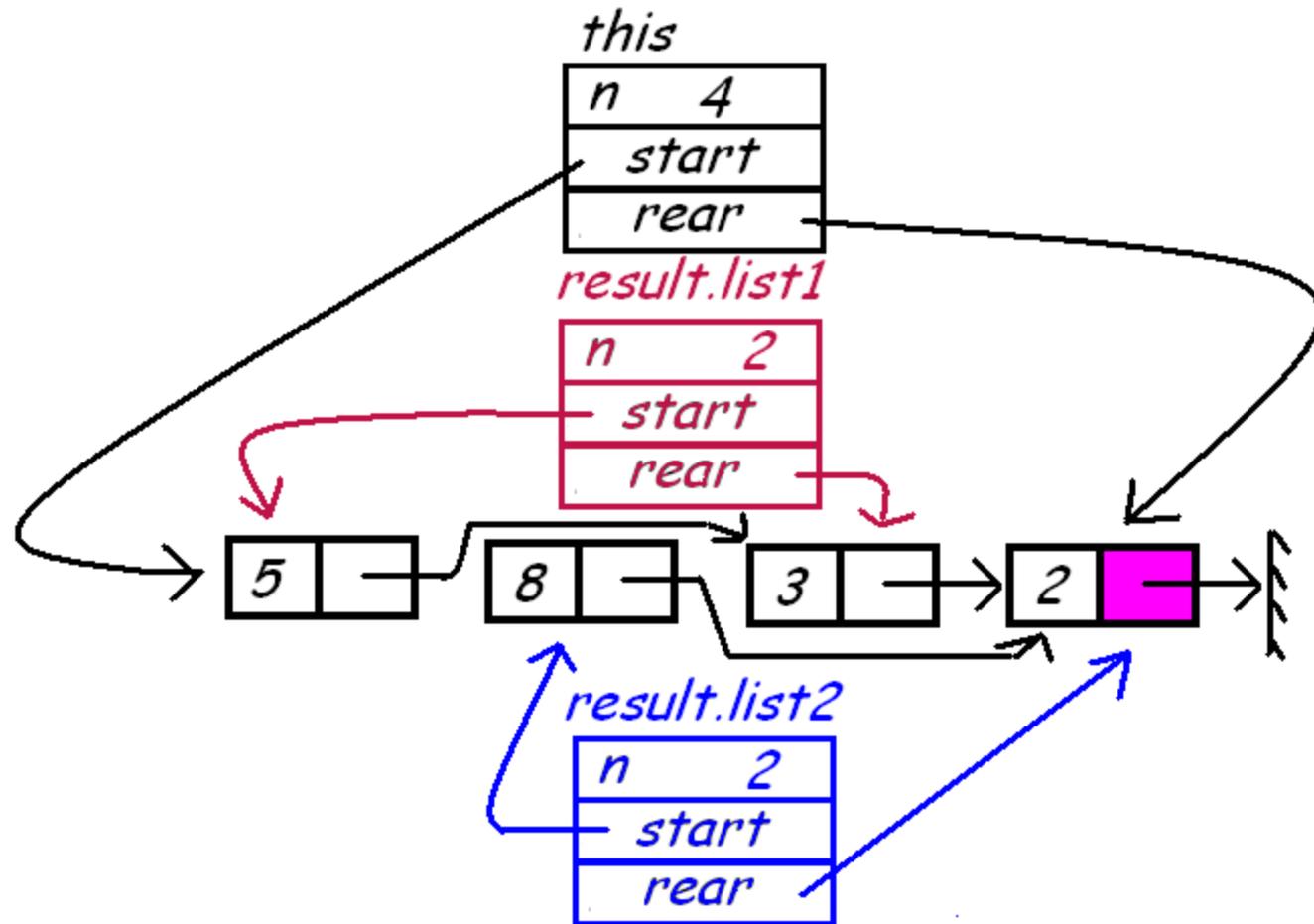
```
result.list1.rear = result.list2.rear.next;
result.list1.n++;
```



`result.list2.rear.next = result.list1.rear.next;`



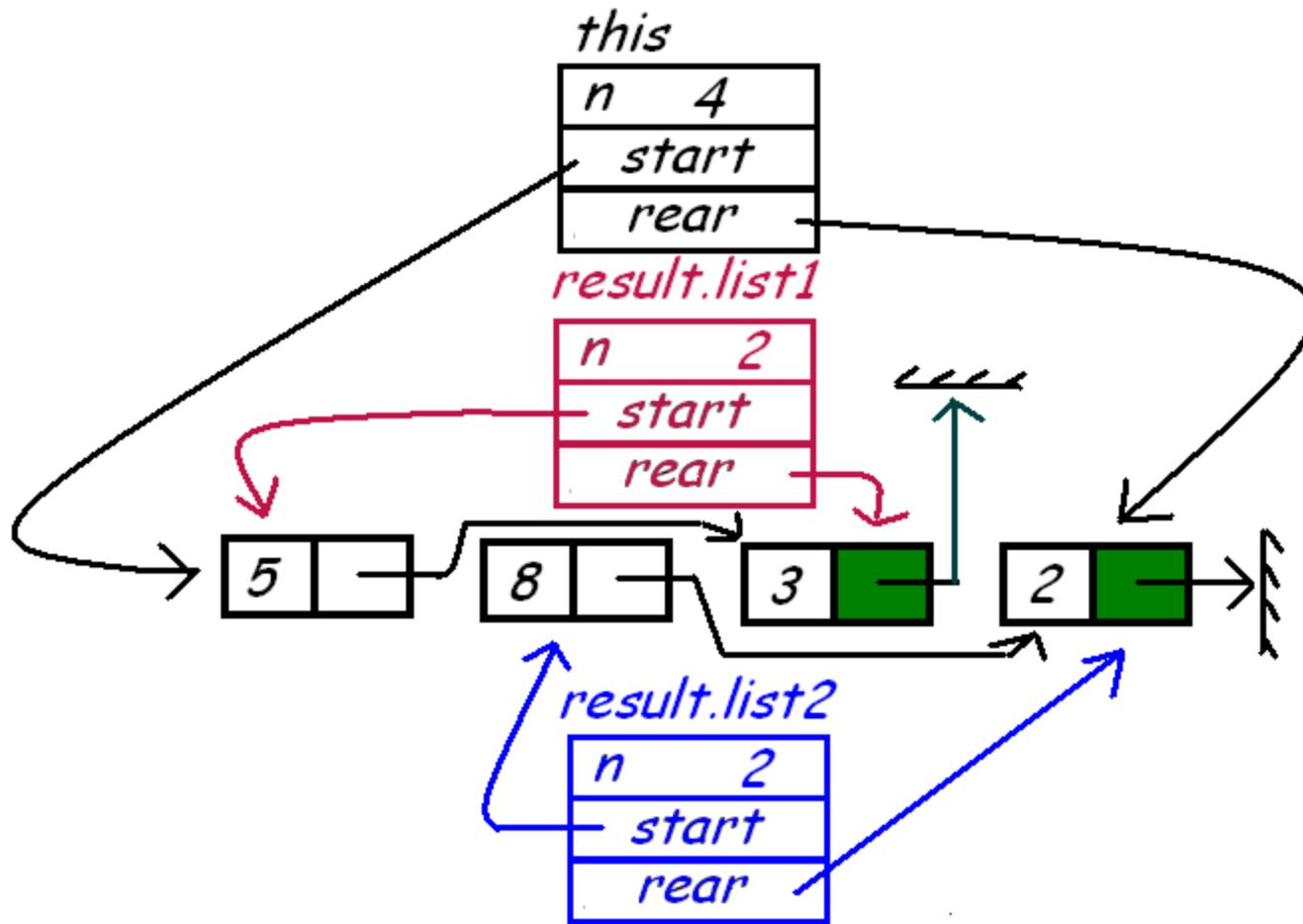
`result.list2.rear = result.list1.rear.next;`



result.list2.n++;

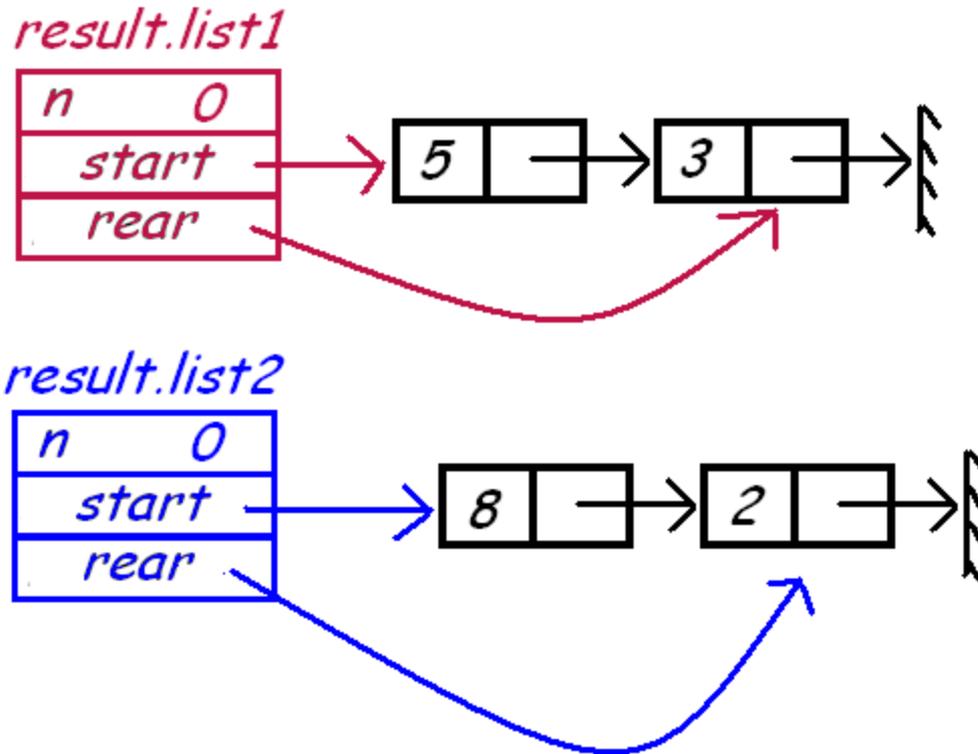
The while loop now terminates:

while (result.list2.rear.next != null)



```
result.list1.rear.next= null;
result.list2.rear.next= null;
```

Redrawing the picture:



`return(result);`

The code I gave you is buggy. Try this example to see what goes wrong. Fix the program so it works as it should.

