Dmitri Stanchevici
Unit 4
Module 1

Ex. 1.1

A = {1, 3, 5, 7, 9, 11, 13, 15, 17, 19}
B = {1, 2, 3, 5, 7, 11, 13, 17, 19}
C = {1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29}
D = {3, 5, 7}
E = {2, 3}

Solve:
(((A intersection B) – D) unition E) intersection C)

1. ((({1, 2, 5, 7, 11, 13, 17, 19}) – D) unition E) intersection C)
2. ((({1, 2, 5, 7, 11, 13, 17, 19}) – {3, 5, 7}) unition E) intersection C)
3. (({1, 2, 11, 13, 17, 19} unition {2, 3}) intersection C)
4. ({1, 2, 3, 11, 13, 17, 19} intersection {1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29})
5. {1, 2, 3, 11, 13, 17, 19}

Result:
{1, 2, 3, 11, 13, 17, 19}

Ex. 1.8

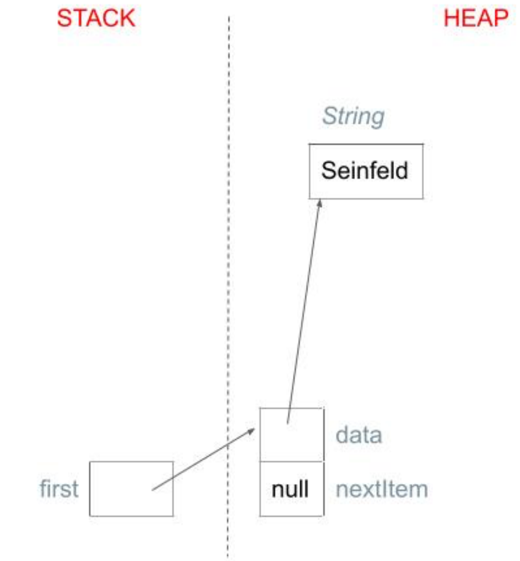
The signature of the contains method should be

```
public boolean contains (String s)
{
        // …
}
```

Ex. 1.12

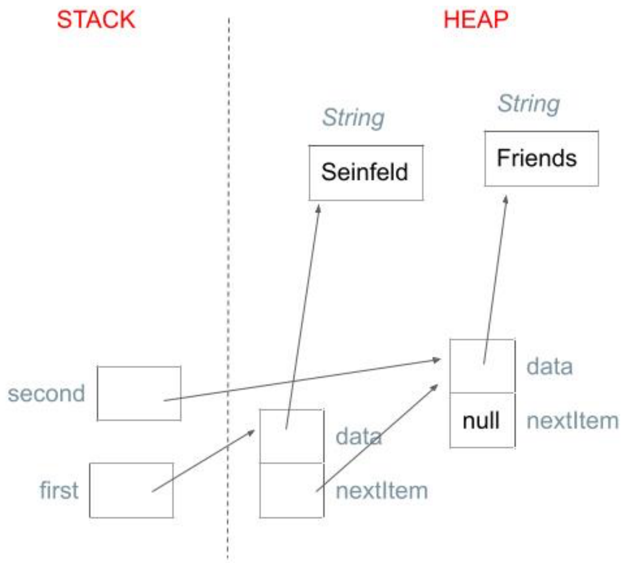
1

```
ListItem first = new ListItem();
first.data = "Seinfeld";
```



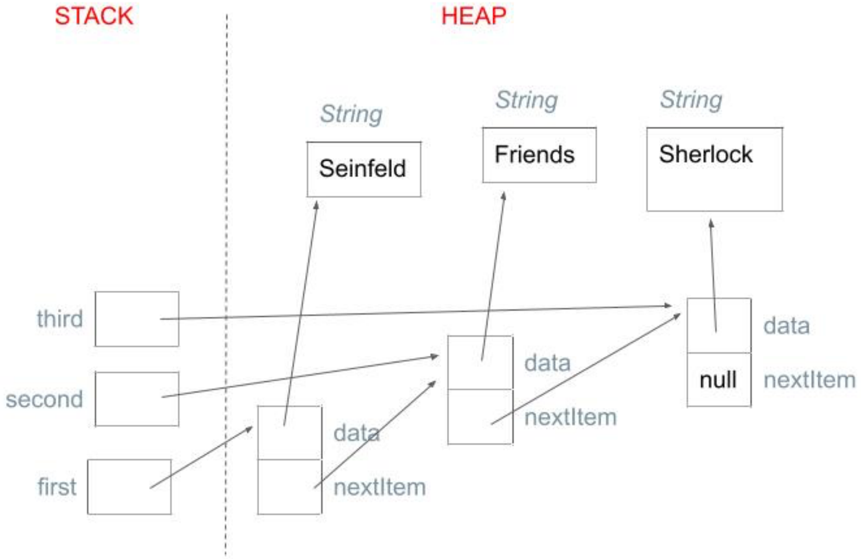
2

```
ListItem second = new ListItem();
second.data = "Friends";
first.next = second;
```



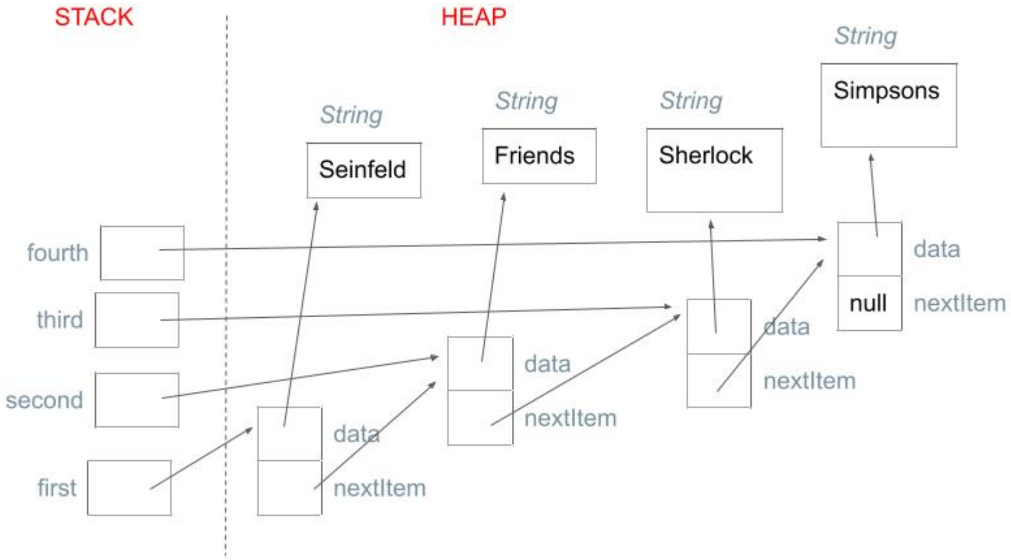
3

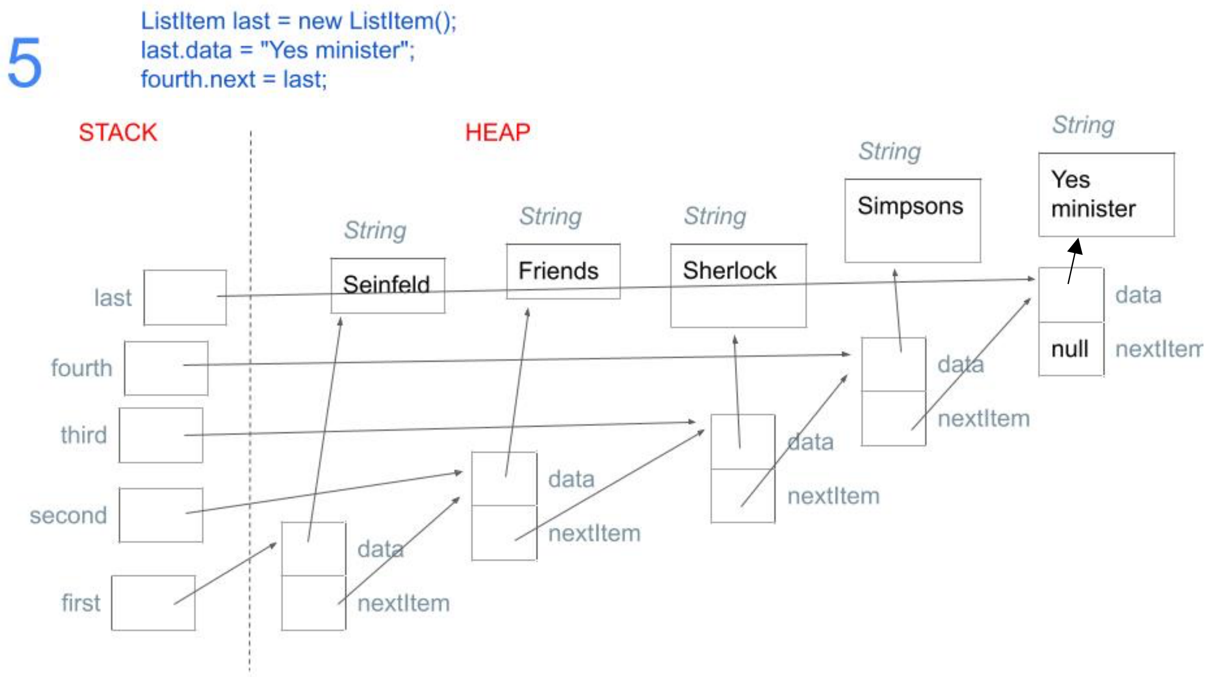
```
ListItem third = new ListItem();
third.data = "Sherlock";
second.next = third;
```



4

```
ListItem fourth = new ListItem();
fourth.data = "Simpsons";
third.next = fourth;
```



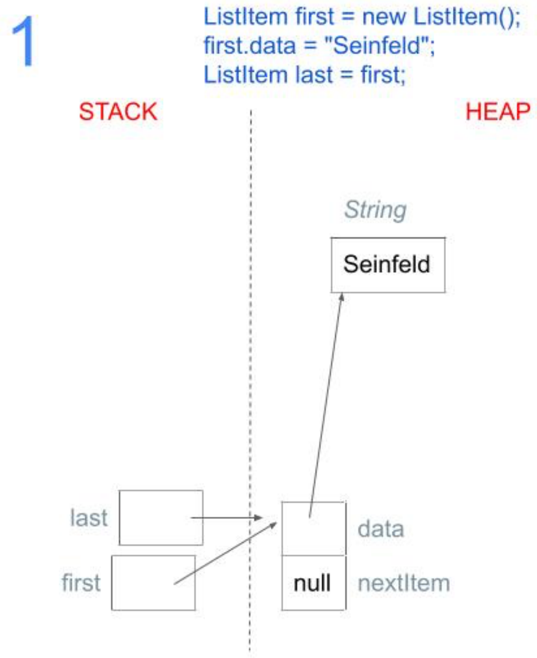


5

```
ListItem last = new ListItem();
last.data = "Yes minister";
fourth.next = last;
```

Ex. 1.13

Memory diagram as new ListItem objects are created.

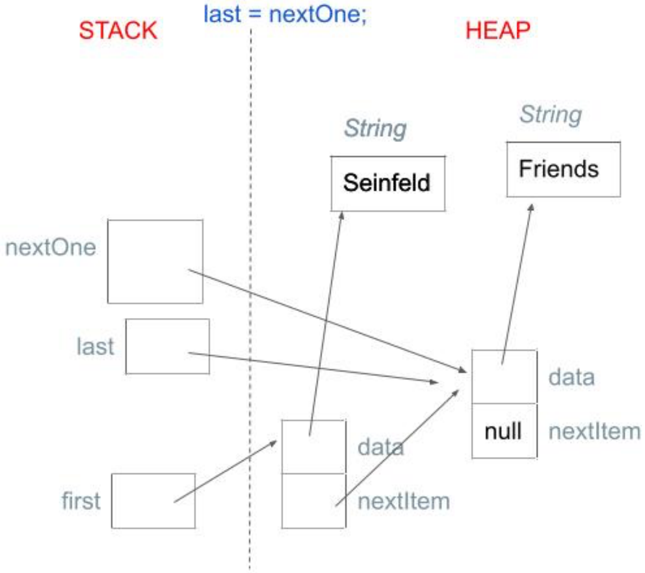


1

```
ListItem first = new ListItem();
first.data = "Seinfeld";
ListItem last = first;
```

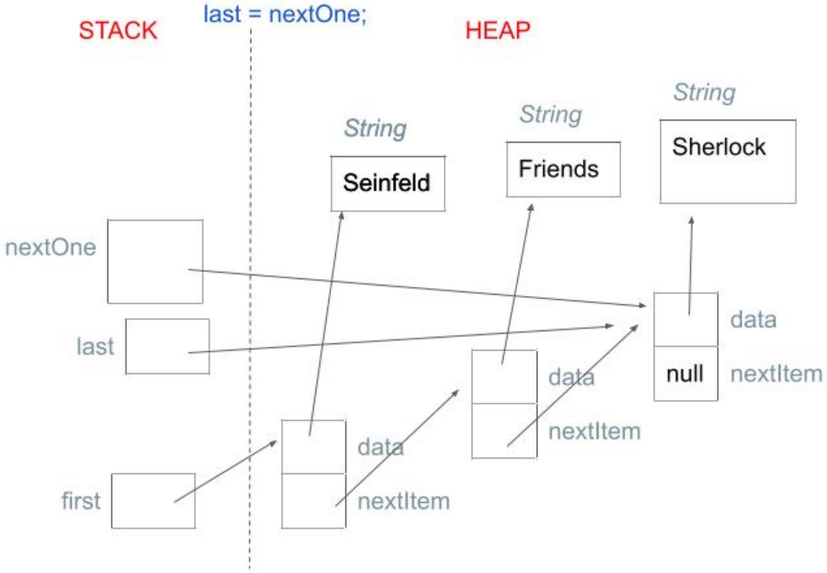
2

```
ListItem nextOne = new ListItem();
nextOne.data = "Friends";
last.next = nextOne;
last = nextOne;
```



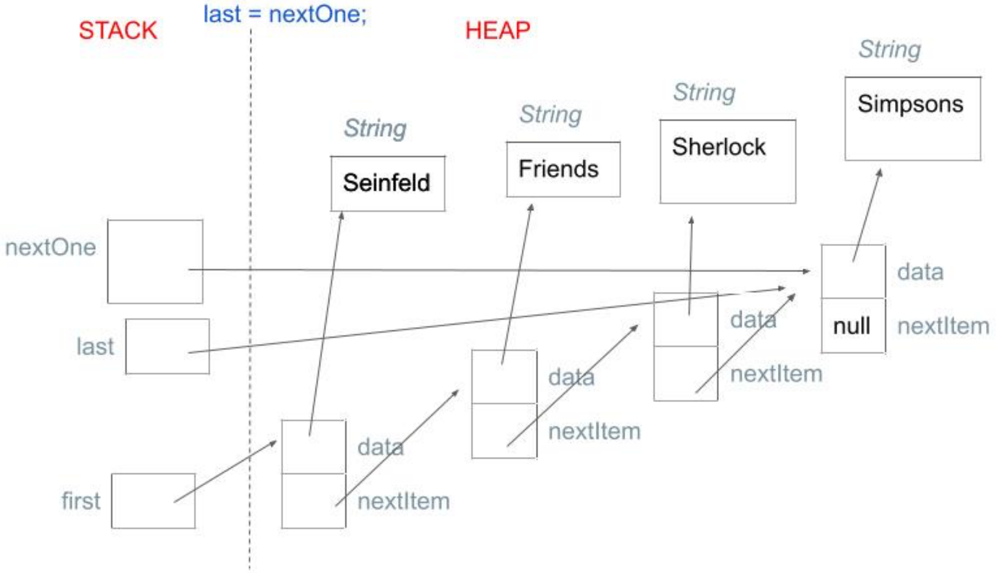
3

```
nextOne = new ListItem();
nextOne.data = "Sherlock";
last.next = nextOne;
last = nextOne;
```



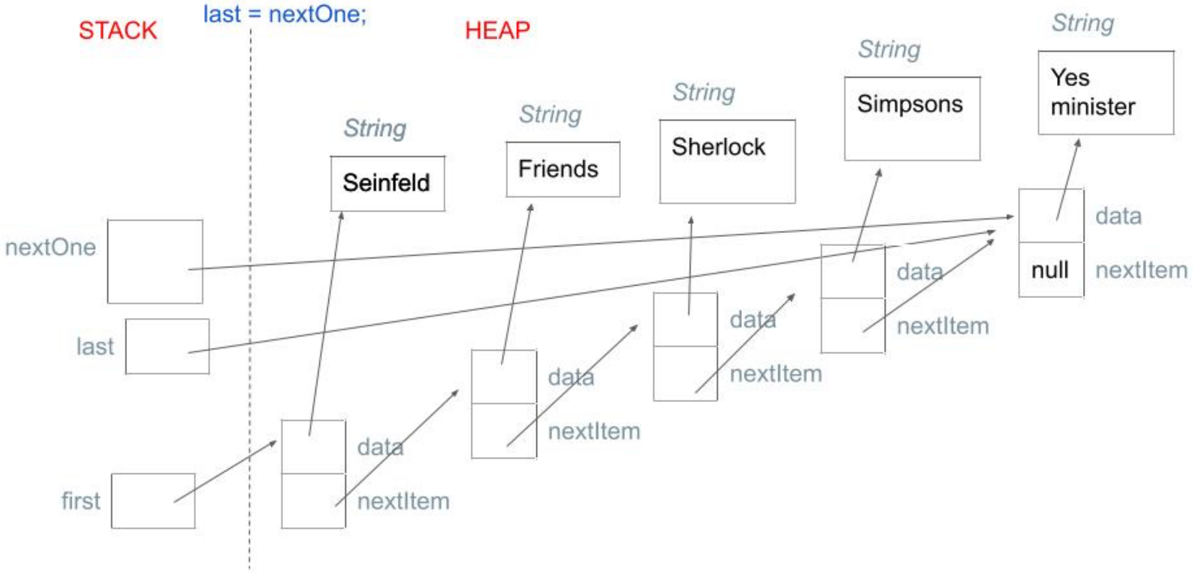
4

```
nextOne = new ListItem();
nextOne.data = "Simpsons";
last.next = nextOne;
last = nextOne;
```



5

```
nextOne = new ListItem();
nextOne.data = "Yes minister";
last.next = nextOne;
last = nextOne;
```

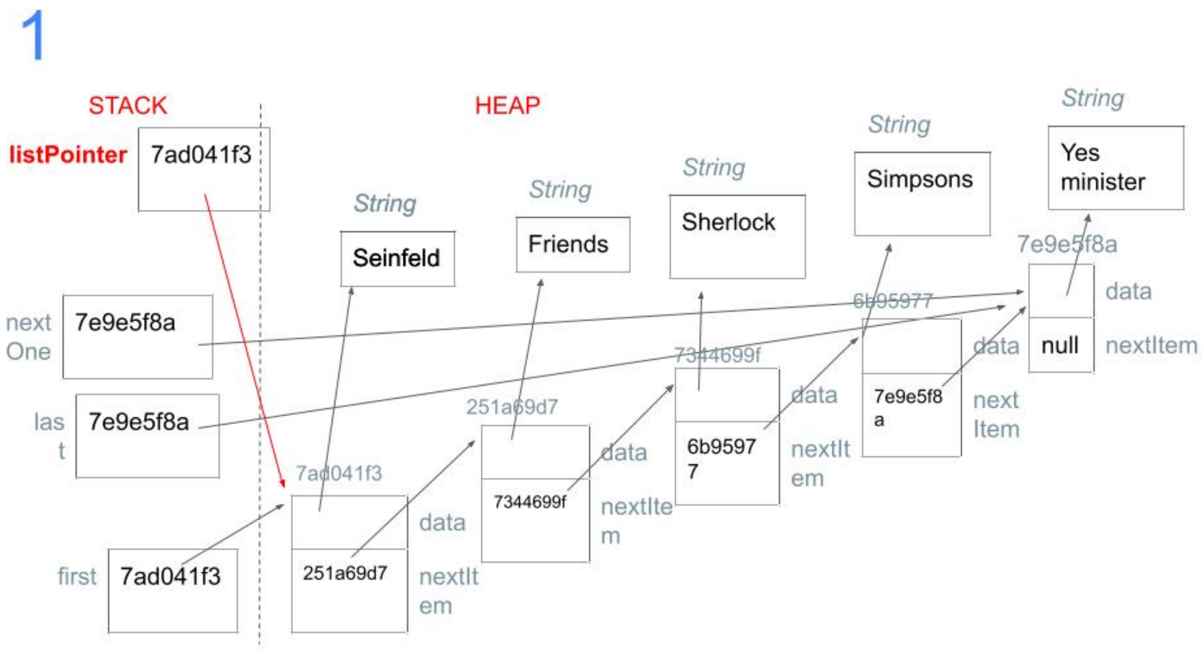


Tracing the values of listPointer as items are printed:

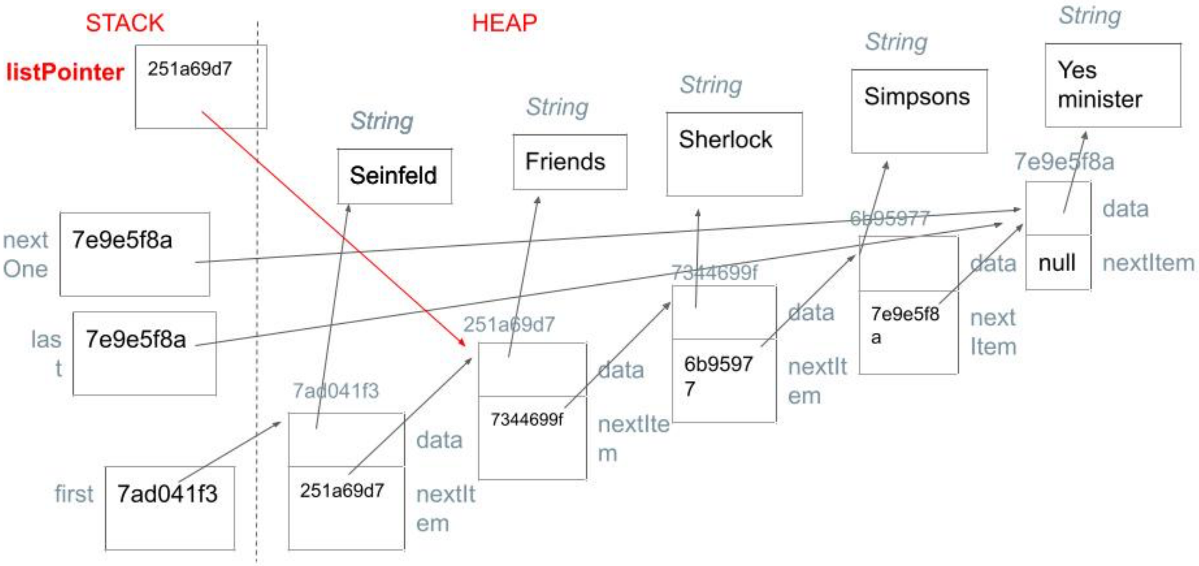
1. listPointer = first
2. listPointer = first.next
3. listPointer = first.next.next
4. listPointer = first.next.next.next
5. listPointer = first.next.next.next.next

Ex. 1.15

Draw list with actual memory addresses and the changing values of listPointer.

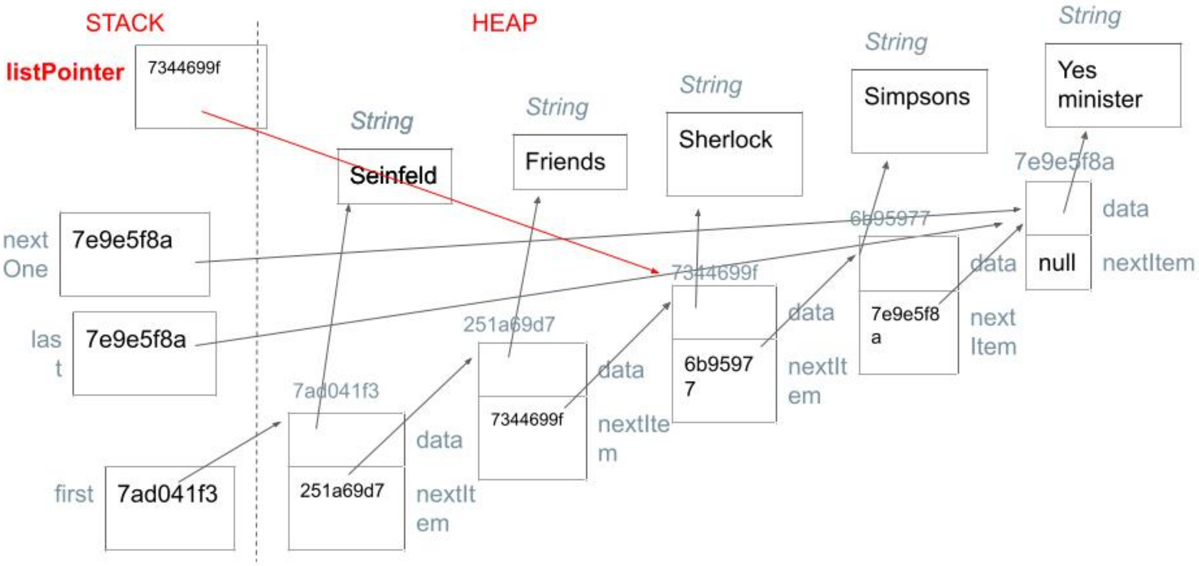


2



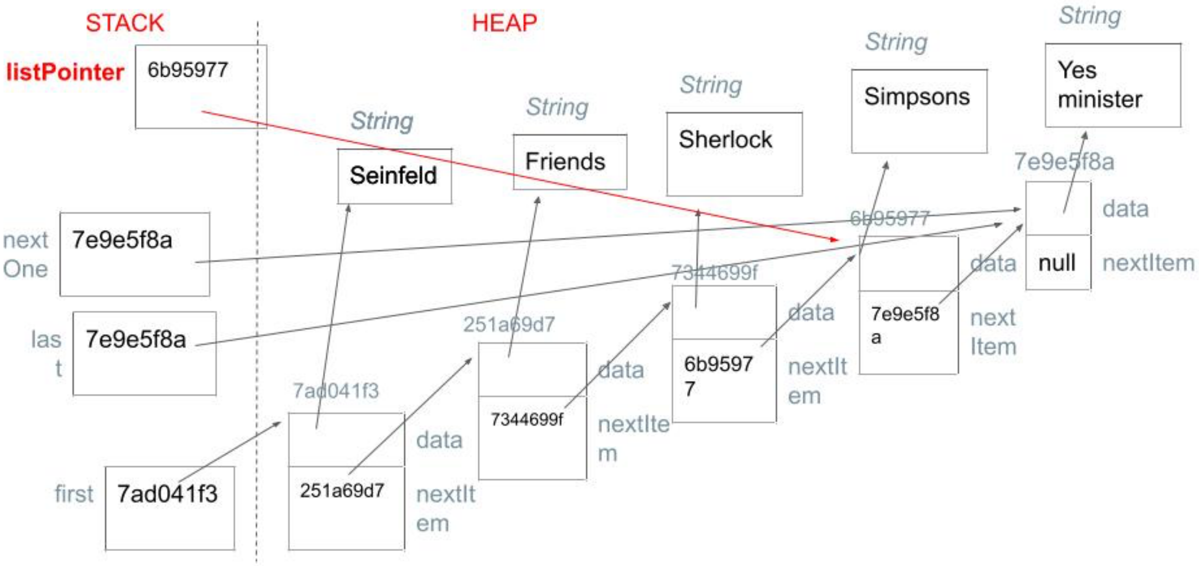
STACK
HEAP
listPointer
251a69d7
next One
7e9e5f8a
las t
7e9e5f8a
first
7ad041f3
String
Seinfeld
String
Friends
String
Sherlock
String
Simpsons
String
Yes minister
7ad041f3
251a69d7
data
nextIt em
251a69d7
7344699f
data
nextIte m
7344699f
6b9597 7
data
nextIt em
6b95977
7e9e5f8 a
data
next Item
7e9e5f8a
null
data
nextItem

3

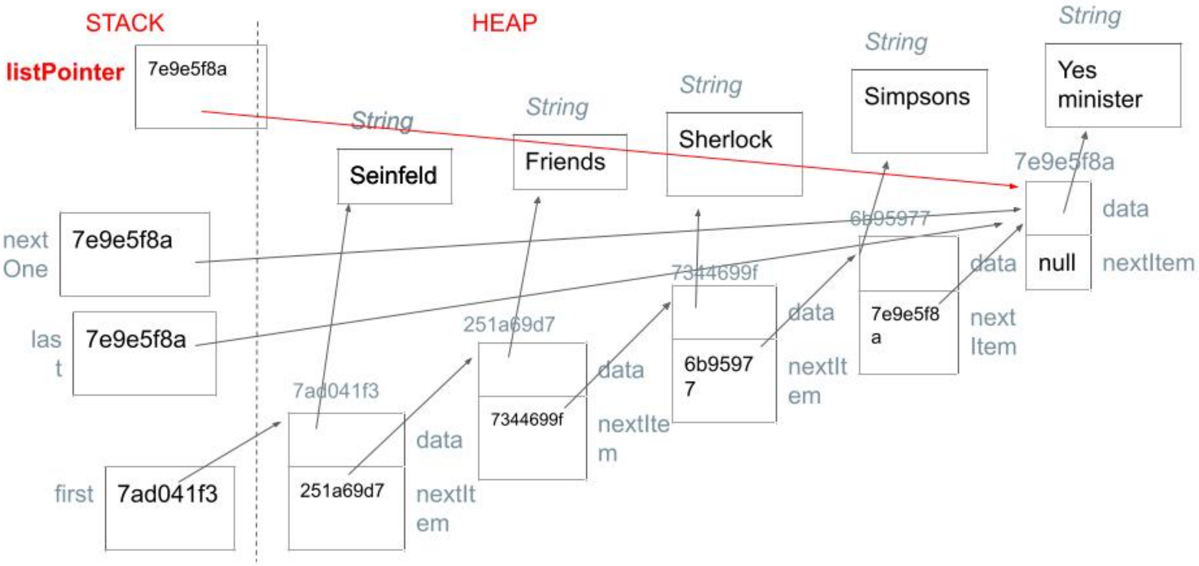


STACK
HEAP
listPointer
7344699f
next One
7e9e5f8a
las t
7e9e5f8a
first
7ad041f3
String
Seinfeld
String
Friends
String
Sherlock
String
Simpsons
String
Yes minister
7ad041f3
251a69d7
data
nextIt em
251a69d7
7344699f
data
nextIte m
7344699f
6b9597 7
data
nextIt em
6b95977
7e9e5f8 a
data
next Item
7e9e5f8a
null
data
nextItem

4

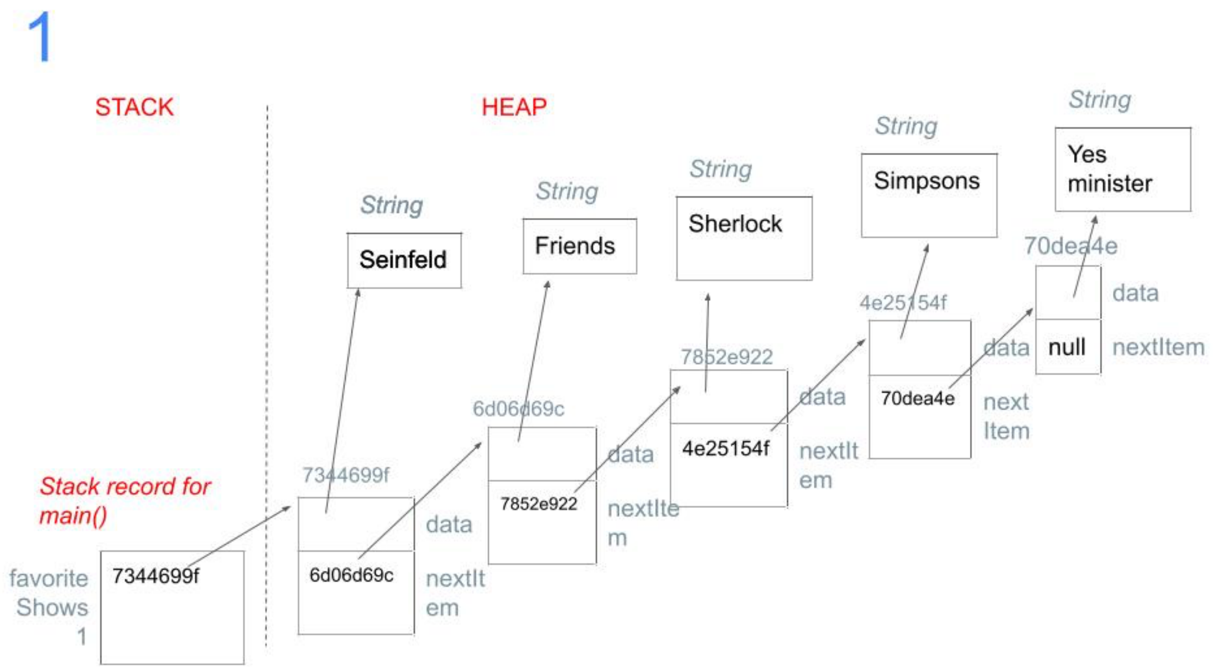

STACK
HEAP
listPointer
6b95977
String
Seinfeld
String
Friends
String
Sherlock
String
Simpsons
String
Yes minister
7e9e5f8a
data
null
nextItem
next One
7e9e5f8a
6b95977
7344699f
data
7e9e5f8a
next Item
las t
7e9e5f8a
251a69d7
data
6b9597 7
nextIt em
7ad041f3
7344699f
data
nextIte m
first
7ad041f3
251a69d7
data
nextIt em

5


STACK
HEAP
listPointer
7e9e5f8a
String
Seinfeld
String
Friends
String
Sherlock
String
Simpsons
String
Yes minister
7e9e5f8a
data
null
nextItem
next One
7e9e5f8a
6b95977
7344699f
data
7e9e5f8a
next Item
las t
7e9e5f8a
251a69d7
data
6b9597 7
nextIt em
7ad041f3
7344699f
data
nextIte m
first
7ad041f3
251a69d7
data
nextIt em

Ex. 1.19

Draw the memory picture with the addresses that are printed.


1
STACK
HEAP
Stack record for main()
favorite Shows 1
7344699f
String
Seinfeld
String
Friends
String
Sherlock
String
Simpsons
String
Yes minister
7344699f
data
6d06d69c
nextItem
6d06d69c
data
7852e922
nextItem
7852e922
data
4e25154f
nextItem
4e25154f
data
70dea4e
next Item
70dea4e
data
null
nextItem