Fiona O’Connell
Ayushi Sharma

Data Structures Lab 8

Running the file:

- The makefile should allow everything to be compiled and run with just

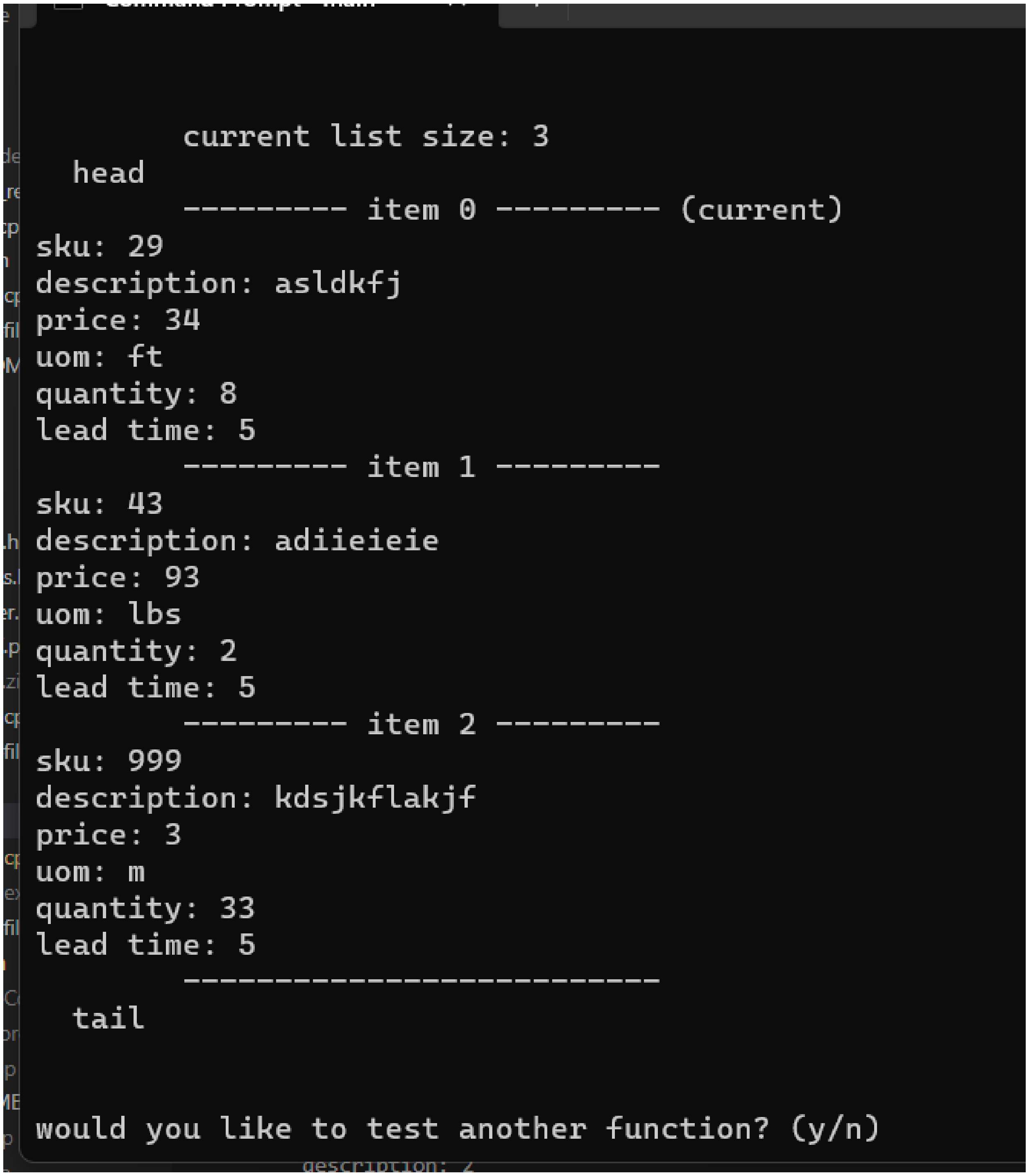
make
main

Objectives and concepts

For this lab we created an ordered linked list using a template class and a separate class for the items to add to the list. We created a separate node structure to hold a pointer to the class item and the pointers to next and previous items in the list. The main function tests each function in the linked list class by allowing a user to pick which member function to test, then inputting any parameters. After each individual test, it displays the full current contents of the list to the screen.

Linked lists and arrays each have different uses but one of the advantages of using linked lists is that they can be far easier to modify while staying ordered. This makes linked lists (and pointers in general) an important concept to understand for anyone going into computer science.

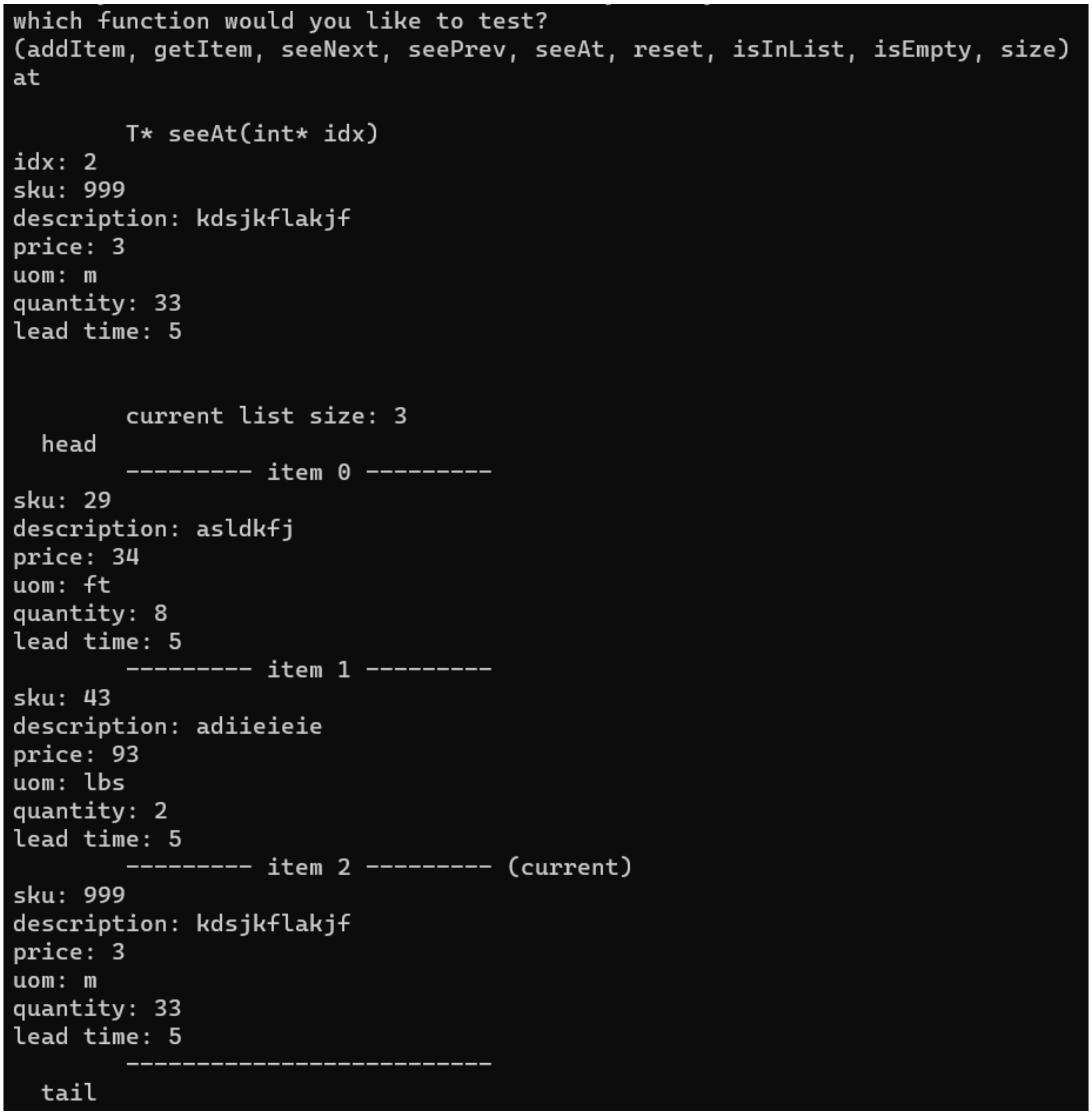
Test screenshots



```
        current list size: 3
  head
        --------- item 0 --------- (current)
sku: 29
description: asldkfj
price: 34
uom: ft
quantity: 8
lead time: 5
        --------- item 1 ---------
sku: 43
description: adiieieie
price: 93
uom: lbs
quantity: 2
lead time: 5
        --------- item 2 ---------
sku: 999
description: kdsjkflakjf
price: 3
uom: m
quantity: 33
lead time: 5
        --------------------------
  tail


would you like to test another function? (y/n)
```

(after adding three items)

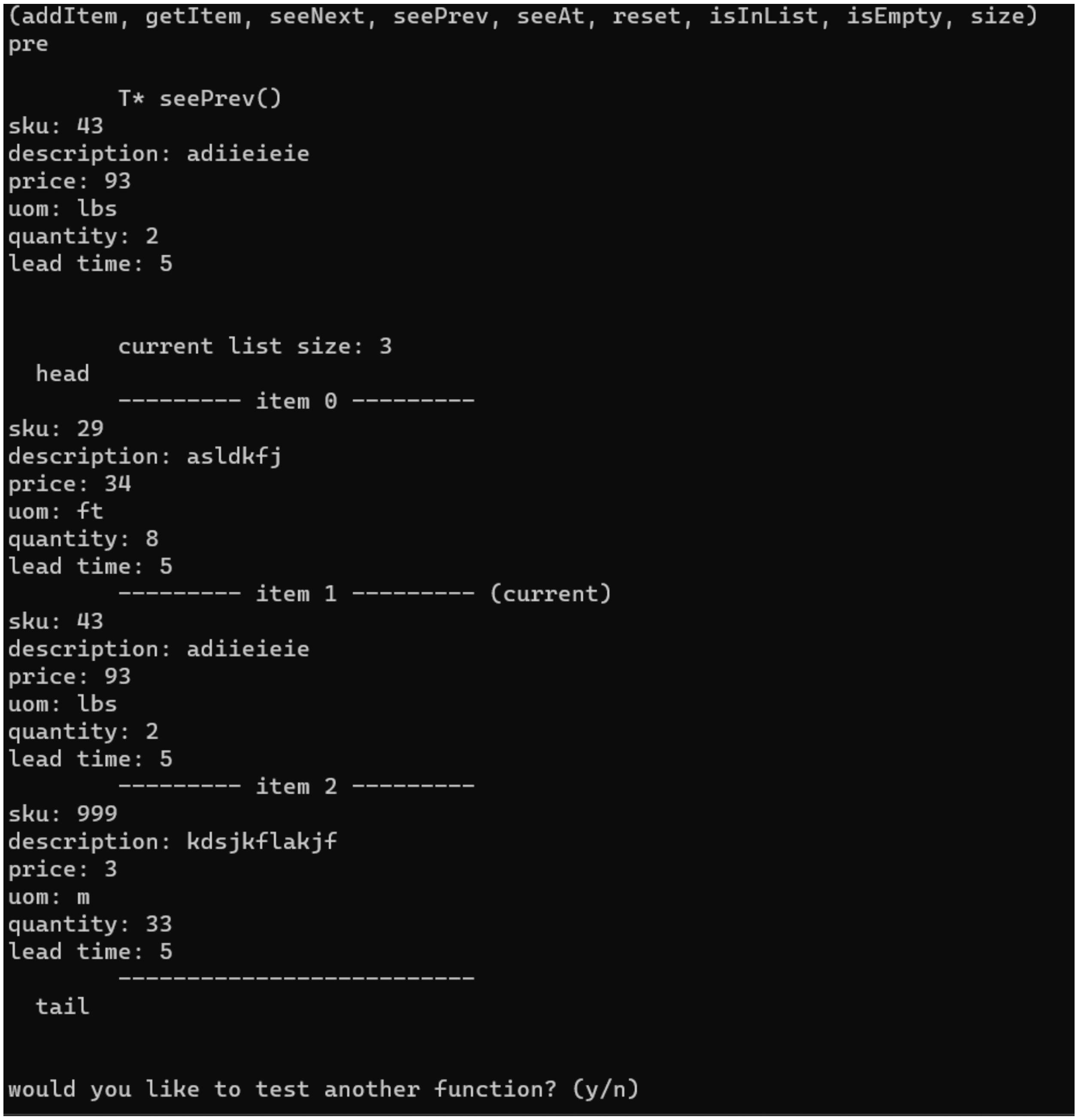


```
which function would you like to test?
(addItem, getItem, seeNext, seePrev, seeAt, reset, isInList, isEmpty, size)
at

        T* seeAt(int* idx)
idx: 2
sku: 999
description: kdsjkflakjf
price: 3
uom: m
quantity: 33
lead time: 5


        current list size: 3
  head
        --------- item 0 ---------
sku: 29
description: asldkfj
price: 34
uom: ft
quantity: 8
lead time: 5
        --------- item 1 ---------
sku: 43
description: adiieieie
price: 93
uom: lbs
quantity: 2
lead time: 5
        --------- item 2 --------- (current)
sku: 999
description: kdsjkflakjf
price: 3
uom: m
quantity: 33
lead time: 5
        --------------------------
  tail
```

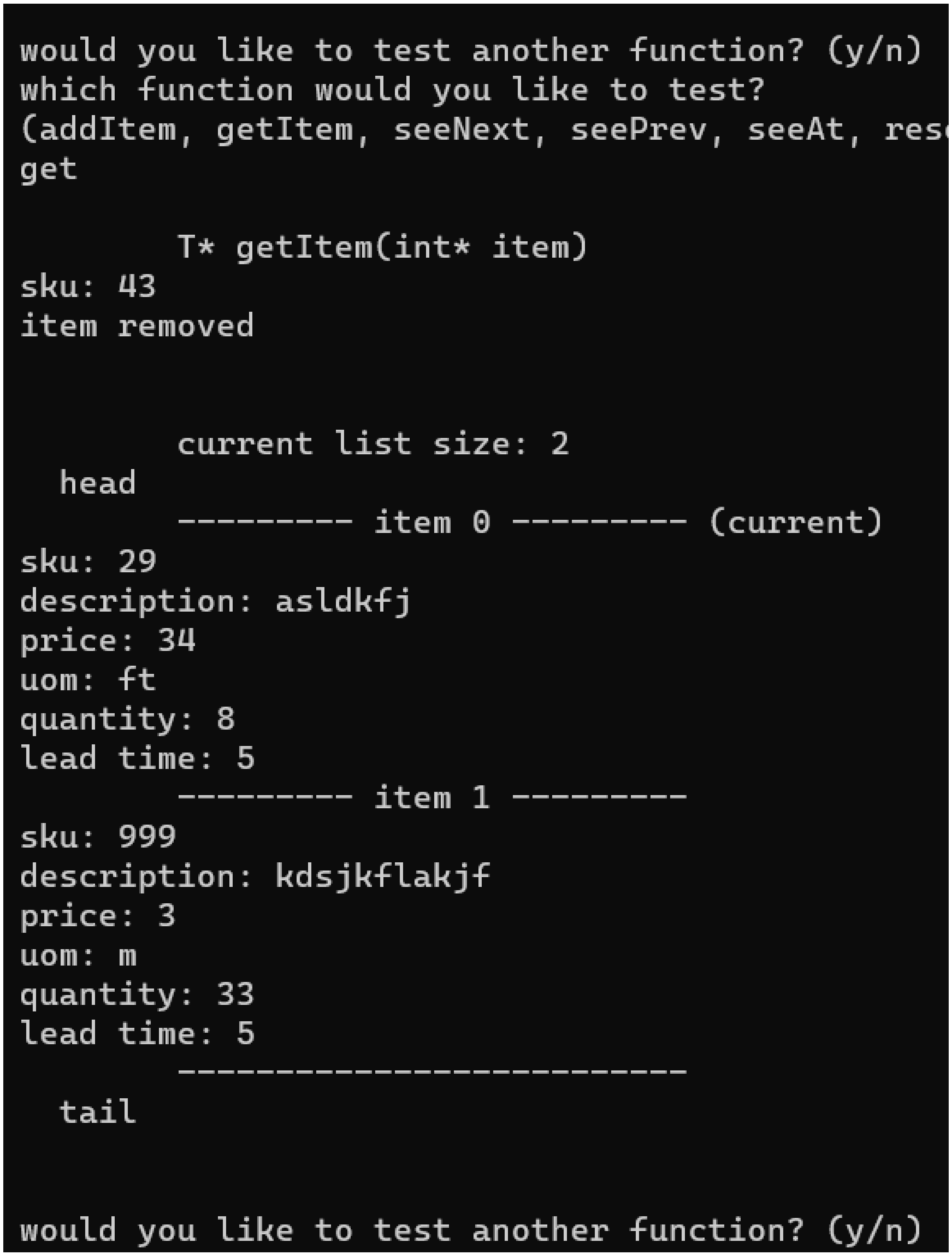
(seeAt(2))


```
(addItem, getItem, seeNext, seePrev, seeAt, reset, isInList, isEmpty, size)
pre

        T* seePrev()
sku: 43
description: adiieieie
price: 93
uom: lbs
quantity: 2
lead time: 5


        current list size: 3
  head
        --------- item 0 ---------
sku: 29
description: asldkfj
price: 34
uom: ft
quantity: 8
lead time: 5
        --------- item 1 --------- (current)
sku: 43
description: adiieieie
price: 93
uom: lbs
quantity: 2
lead time: 5
        --------- item 2 ---------
sku: 999
description: kdsjkflakjf
price: 3
uom: m
quantity: 33
lead time: 5
        --------------------------
  tail


would you like to test another function? (y/n)
```
(seePrev())



```
would you like to test another function? (y/n)
which function would you like to test?
(addItem, getItem, seeNext, seePrev, seeAt, res
get

        T* getItem(int* item)
sku: 43
item removed


        current list size: 2
  head
        --------- item 0 --------- (current)
sku: 29
description: asldkfj
price: 34
uom: ft
quantity: 8
lead time: 5
        --------- item 1 ---------
sku: 999
description: kdsjkflakjf
price: 3
uom: m
quantity: 33
lead time: 5
        --------------------------
  tail


would you like to test another function? (y/n)
```

(removed 2nd item)