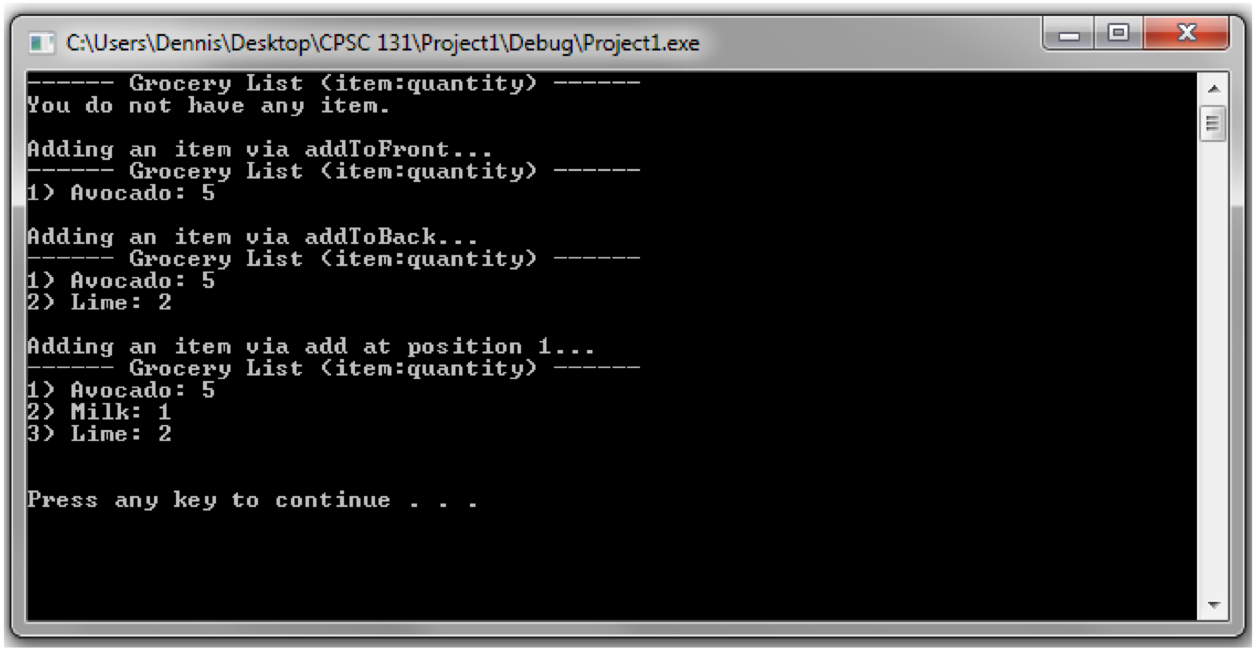
Name: Dennis Wu and Barry Ibarra
Project: Programming Assignment #1
Course: CPSC131
Professor: Dr Abhishek Verma

Pseudo code for Add function

```
Function: add(string itemName, real quantity, integer position)
Input:  itemName, quantity, position
Return: nothing
   INIT new item node with itemName, quantity
   IF list is empty OR position is 0
   THEN
         CALL addToFront with itemName, quantity
   ELSE IF position >= total count of items
   THEN
         CALL addToBack with itemName, quantity
   ELSE
         INIT current = head
         FOR i =1 step 1 to position
               CALL goNext with current
         END FOR
         SET new previous item = previous item
         SET new next item = current item
         SET previous next item = new item
         SET previous current item = new item
   END IF
   INCREMENT itemCount
```

Screenshot of add and print backward function

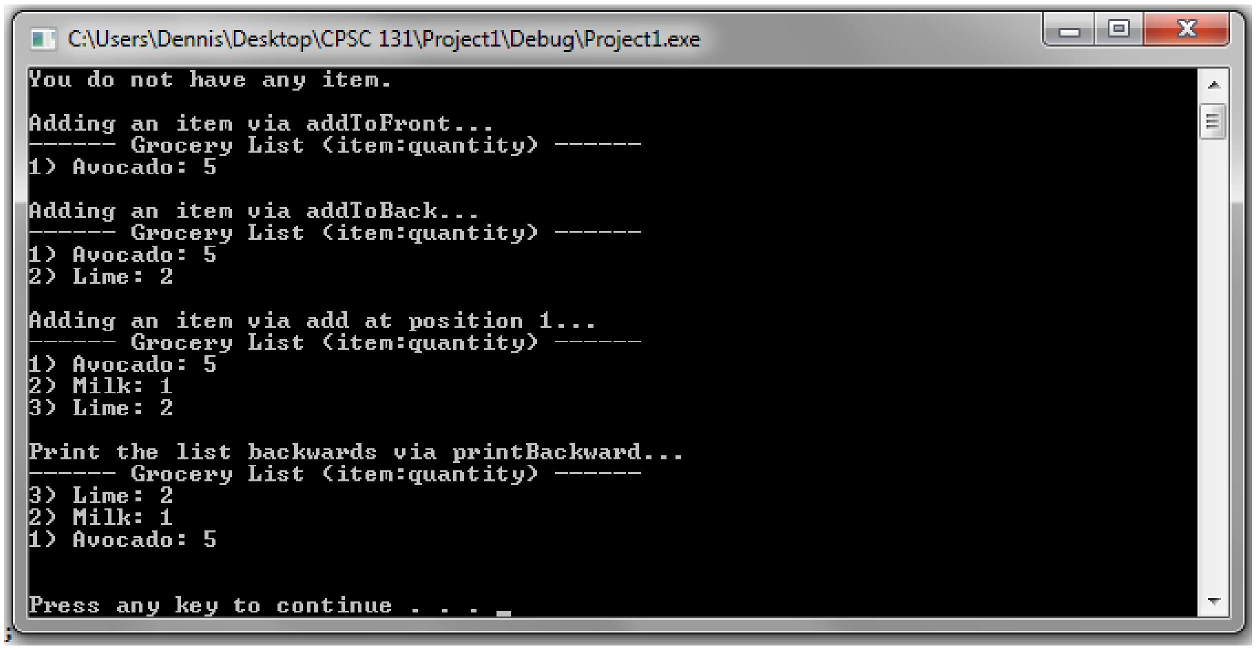
void add(string itemName, double quantity, int position)



C:\Users\Dennis\Desktop\CPSC 131\Project1\Debug\Project1.exe

```
------ Grocery List (item:quantity) ------
You do not have any item.

Adding an item via addToFront...
------ Grocery List (item:quantity) ------
1) Avocado: 5

Adding an item via addToBack...
------ Grocery List (item:quantity) ------
1) Avocado: 5
2) Lime: 2

Adding an item via add at position 1...
------ Grocery List (item:quantity) ------
1) Avocado: 5
2) Milk: 1
3) Lime: 2


Press any key to continue . . .
```

void printBackward() const;



C:\Users\Dennis\Desktop\CPSC 131\Project1\Debug\Project1.exe

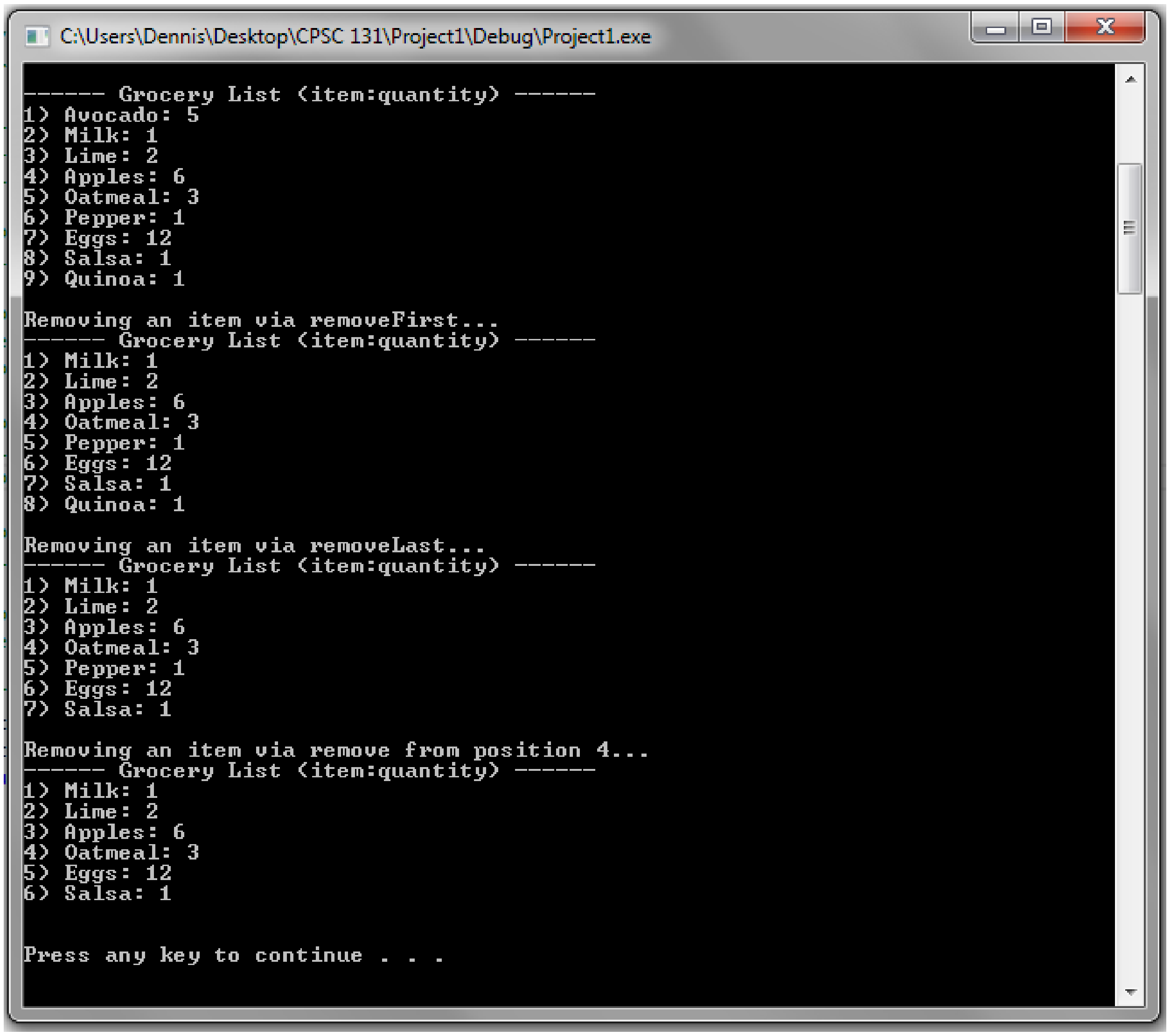
```
You do not have any item.

Adding an item via addToFront...
------ Grocery List (item:quantity) ------
1) Avocado: 5

Adding an item via addToBack...
------ Grocery List (item:quantity) ------
1) Avocado: 5
2) Lime: 2

Adding an item via add at position 1...
------ Grocery List (item:quantity) ------
1) Avocado: 5
2) Milk: 1
3) Lime: 2

Print the list backwards via printBackward...
------ Grocery List (item:quantity) ------
3) Lime: 2
2) Milk: 1
1) Avocado: 5


Press any key to continue . . . _
```

Pseudo code for remove function

```
Function: remove(integer position)
Input:  position
Return: nothing
  INIT lastItem to itemCount - 1
  IF list <> empty AND position < itemCount
  THEN
        INIT current = head
        IF position is 0 AND itemCount is 1
        THEN
                SET head, tail to NULL
        ELSE
                IF position is 0
                THEN
                        CALL removeFirst
                ELSE IF position is lastItem
                        CALL removeLast
                ELSE
                        FOR  i = 1 step 1 to position
                        CALL goNext with current
                        END FOR
                        SET previous next item = next current item
                END IF
        END IF
  END IF
  DECREMENT itemCount
```

Screenshot of remove function

void remove(int position)



C:\Users\Dennis\Desktop\CPSC 131\Project1\Debug\Project1.exe

```
------ Grocery List (item:quantity) ------
1) Avocado: 5
2) Milk: 1
3) Lime: 2
4) Apples: 6
5) Oatmeal: 3
6) Pepper: 1
7) Eggs: 12
8) Salsa: 1
9) Quinoa: 1

Removing an item via removeFirst...
------ Grocery List (item:quantity) ------
1) Milk: 1
2) Lime: 2
3) Apples: 6
4) Oatmeal: 3
5) Pepper: 1
6) Eggs: 12
7) Salsa: 1
8) Quinoa: 1

Removing an item via removeLast...
------ Grocery List (item:quantity) ------
1) Milk: 1
2) Lime: 2
3) Apples: 6
4) Oatmeal: 3
5) Pepper: 1
6) Eggs: 12
7) Salsa: 1

Removing an item via remove from position 4...
------ Grocery List (item:quantity) ------
1) Milk: 1
2) Lime: 2
3) Apples: 6
4) Oatmeal: 3
5) Eggs: 12
6) Salsa: 1


Press any key to continue . . .
```

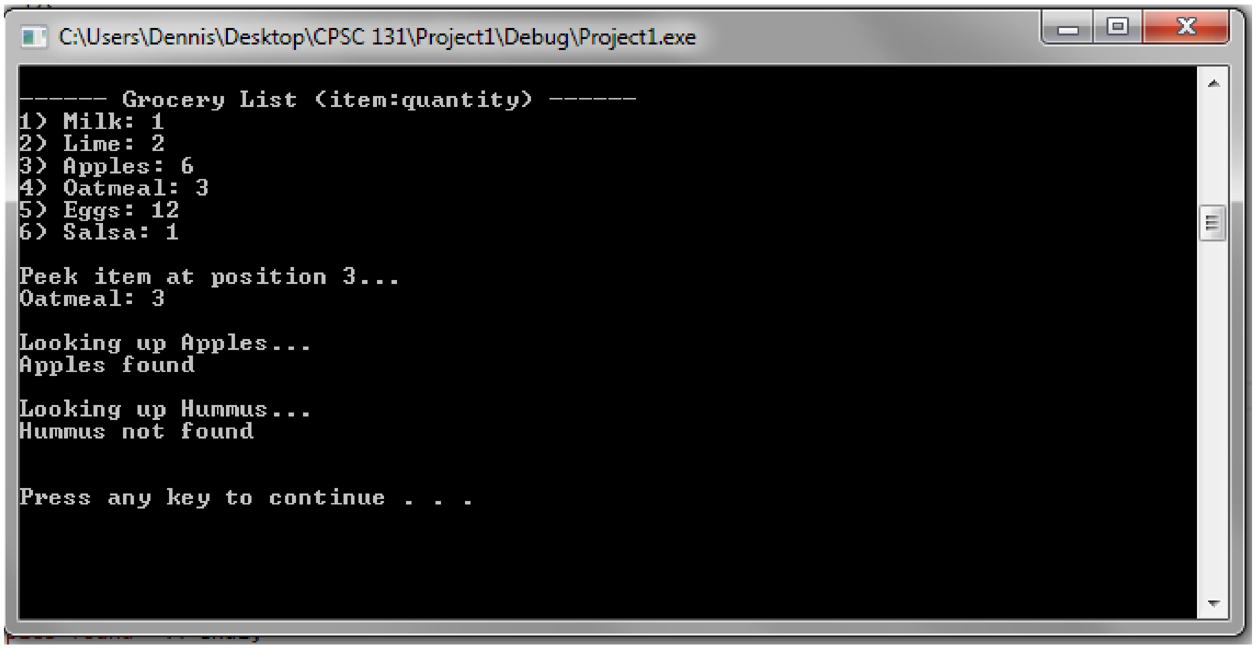
Pseudo code for peek and look up function

```
Function: peek(integer position)
Input:  position
Return: nothing
  IF position < itemCount AND position >= 0
  THEN
        INIT current = head
        FOR i = 1 step 1 to position
                CALL goNext with current
        END FOR
        PRINT itemName, qauntity
  END IF


Function: lookup(string itemName)
Input:  position
Return: true or false
  INIT current = head
  WHILE  current NOT EQUAL empty
  DO
        IF current item name = itemName
        THEN
                lookup <- TRUE
        END IF
        CALL goNext with current
  END WHILE
  lookup <- FALSE
```

Screenshot of peek and look up function

void peek(int pos) const;

bool lookup(string itemName) const;


C:\Users\Dennis\Desktop\CPSC 131\Project1\Debug\Project1.exe

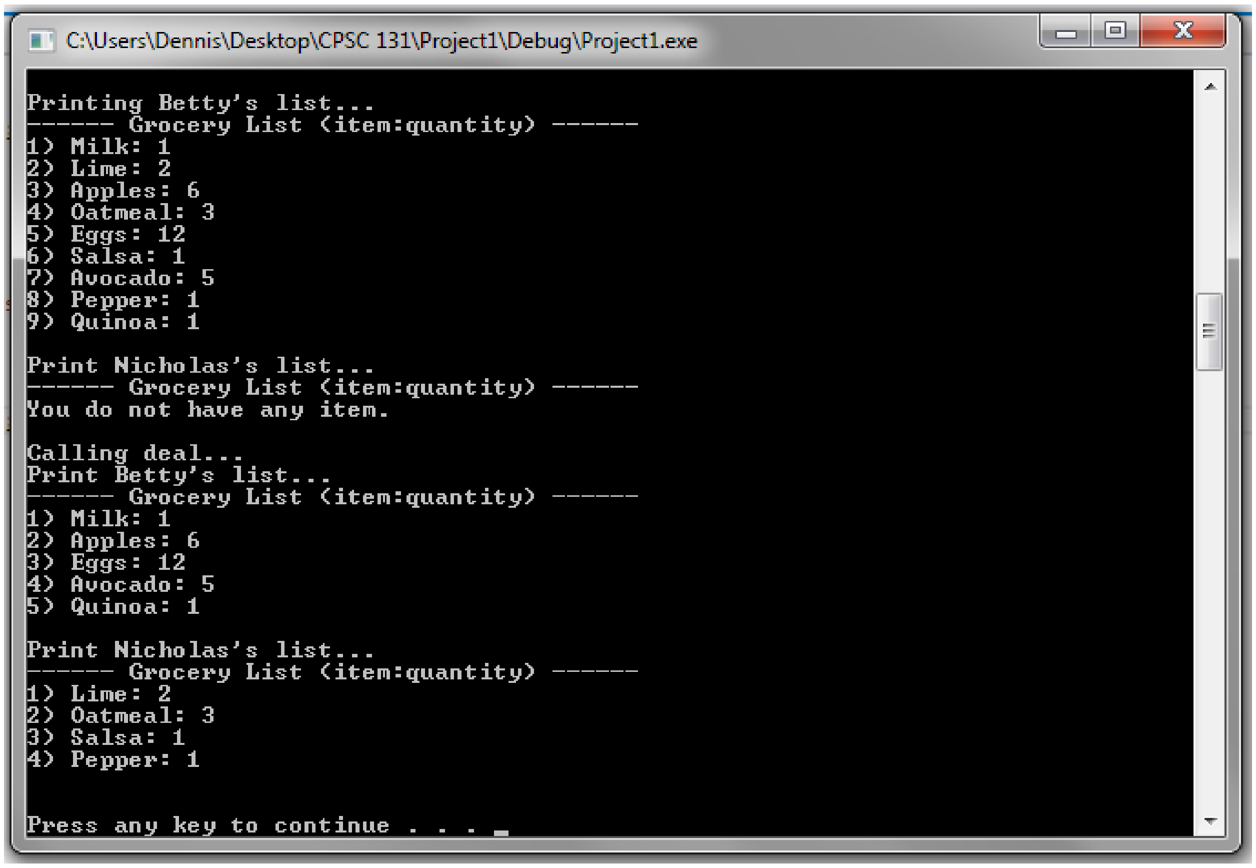
```
------ Grocery List (item:quantity) ------
1) Milk: 1
2) Lime: 2
3) Apples: 6
4) Oatmeal: 3
5) Eggs: 12
6) Salsa: 1

Peek item at position 3...
Oatmeal: 3

Looking up Apples...
Apples found

Looking up Hummus...
Hummus not found


Press any key to continue . . .
```

Pseudo code for deal function

```
Function: deal(GroceryList secondList)
Input:  second grocery list
Return: nothing
   INIT current = head
   INIT numSplit = itemCount/2
   IF itemCount NOT EQUAL 1
   THEN
         IF itemCount is 2
         THEN
               CALL goNext with current
               CALL secondList.addToFront with itemName, quantity
               CALL removeLast
         ELSE
               FOR i = 1 step 1 to numSplit
                     CALL goNext with current
                     CALL secondList.addToBack with itemName, quantity
                     CALL goNext with current
                     CALL remove with i+1
               END FOR
         END IF
    END IF
```

Screenshot of deal function

void deal(GroceryList & secondList)



C:\Users\Dennis\Desktop\CPSC 131\Project1\Debug\Project1.exe

```
Printing Betty's list...
------ Grocery List (item:quantity) ------
1) Milk: 1
2) Lime: 2
3) Apples: 6
4) Oatmeal: 3
5) Eggs: 12
6) Salsa: 1
7) Avocado: 5
8) Pepper: 1
9) Quinoa: 1

Print Nicholas's list...
------ Grocery List (item:quantity) ------
You do not have any item.

Calling deal...
Print Betty's list...
------ Grocery List (item:quantity) ------
1) Milk: 1
2) Apples: 6
3) Eggs: 12
4) Avocado: 5
5) Quinoa: 1

Print Nicholas's list...
------ Grocery List (item:quantity) ------
1) Lime: 2
2) Oatmeal: 3
3) Salsa: 1
4) Pepper: 1


Press any key to continue . . .
```