

Solution to #3

LinkedList.h definitions

```
1 void evenMean();  
2 /*-----  
3 Determine the mean value among the even elements in the list.  
4 Precondition: none  
5 Postcondition:  
6 -----*/  
7  
8 void oddMean();  
9 /*-----  
10 Determine the mean value among the odd elements in the list.  
11 Precondition: none  
12 Postcondition:  
13 -----*/
```

LinkedList.cpp implementation

```
CMakeLists.txt x  LinkedList.h x  LinkedList.cpp x  Linked_List_Tester.cpp x
174 // Definition for the mean in an even length linked list
175 void LinkedList::evenMean() {
176     Node * fastPtr = first;
177     Node * slowPtr = first;
178     if (first != NULL) {
179         while (fastPtr!=NULL && fastPtr->next!=NULL) {
180             fastPtr = fastPtr->next->next;
181             slowPtr= slowPtr->next;
182         }
183         cout << slowPtr->data;
184     }
185 }
186 // Definition for the mean in an odd length linked list
187 void LinkedList::oddMean() {
188     int count = 0;
189     Node *middle = first;
190     while(first!= NULL){
191         if(count%2==1){
192             middle= middle->next;
193         }
194         count++;
195         first=first->next;
196     }
197     if(middle!=NULL){
198         cout << middle->data;
199     }
200 }
201 }
202 }
```

Linked_List_Tester.cpp

```

LinkedList intList;

cout << "\n\nConstructing intList" << endl << endl;

// Test insert()
intList.insert(1, 0);
intList.display(cout);
cout << endl;

intList.insert(2, 1);
intList.display(cout);
cout << endl;

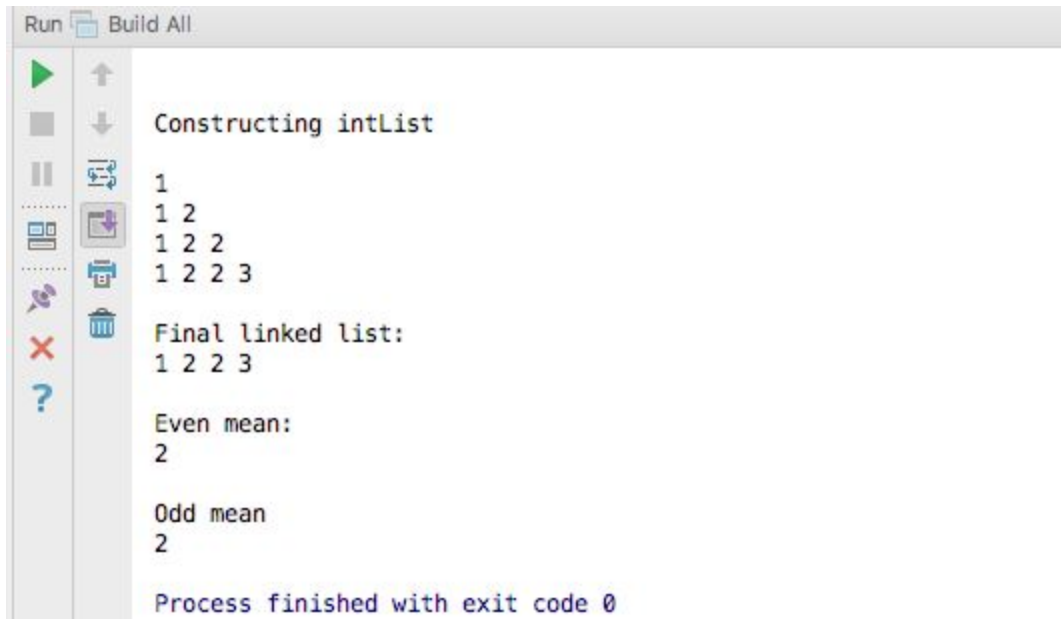
intList.insert(2, 2);
intList.display(cout);
cout << endl;

intList.insert(3, 3);
intList.display(cout);
cout << endl;

cout << "\nFinal linked list: " << endl;
intList.display(cout);
cout << endl;
cout << "\nEven mean: " << endl;
intList.evenMean();
cout << endl;
cout << "\nOdd mean" << endl;
intList.oddMean();
cout << endl;

```

LinkedList OUTPUT



```

Run Build All
Constructing intList
1
1 2
1 2 2
1 2 2 3
Final linked list:
1 2 2 3
Even mean:
2
Odd mean
2
Process finished with exit code 0

```