GTU Department of Computer Engineering

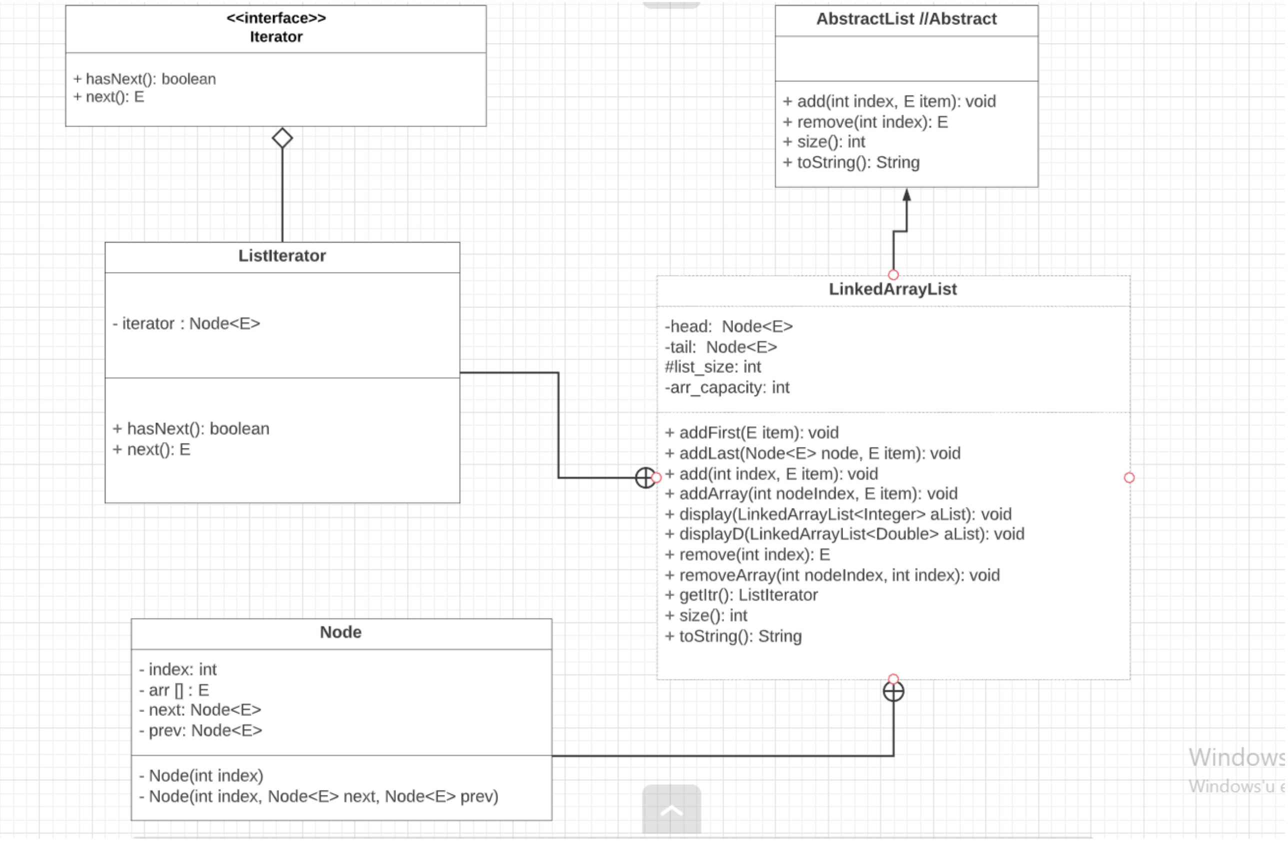
CSE 222/505 - Spring 2020

Homework 3 - Report

Mert Can BEŞİRLİ
1801042663

#Question 1#

1. CLASS DIAGRAMS



2. PROBLEM SOLUTION APPROACH

Our problem is to create nodes in the list and array of data. But arrays integer, double, char etc. can. Therefore, the arrays in the node must be generic type. Node, array can be added to or removed from the node. Array must have a certain maximum capacity. We took 3 in our solution. We can solve the problem in two different ways. Integer and double types. We will have a menu and first integer type transactions then double type transactions. Menu→ A->Adding Node, B->Adding array elements by node index, C->Delete Node, D->Delete array element by node index, E->Display size, F->Display list, Q->Exit first step.

Of course there will be certain scenarios. For example, if we do C without performing A, we will get an error. In other words, it is not possible for us to do any operation on the list without performing transaction A. We also get a note like this in the terminal: Note: LinkedArrayList.java uses unchecked or unsafe operations. Note: Recompile with -Xlint:unchecked for details. We get this note because we did not create an array in generic type because we did an array (E[]) new Object[3]. But this note has no effect on the functioning of the program. It's a legal situation. If the capacity of the current node in the adding array exceeds 3, a new node will be created at the end of the list and the data will be added there.

3. TEST CASES

Test Case ID	Test Scenario	Test Steps	Test Data	Excepted Results	Actual Results	Pass/Fail
T1	Check Add Node with Valid Data	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice:A data=1	List Display Lists 1	As Excepted	Pass
T2	Check Add Node with invalid Data	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by	Choice: A data=1,5	Catch Error Handling then display java.util.InputMismatch Exception	As Expected	Fail

		node index E->Display size F->Display list Q->Exit first step				
T3	Check First Choice B or C or D	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice: B or C or D	First Enter ‘A’ option	As Expected	Fail
T4	Check First Choice Size	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice: E	Size: 0	As Expected	Pass
T5	Check AddArray with valid data after add node	A->Adding Node B->Adding array elements by node	Choice: B ind: 0 data : 5	Lists 1 5	As Expected	Pass

		index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step				
T6	Check AddArray with invalid data after add node	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice: B ind: 12 data: 5	Catch Error Handling java.lang.NullPointer Lists 1	As Expected	Fail
T7	Check AddArray with valid data after add node	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice: C ind: 0	Lists 1 2 Remove first element: 1	As Expected	Pass

T8	Check AddArray with invalid data after add node	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice: C index: 15	Lists 1 5 java.lang.IndexOutOfBoundsException: 15	As Expected	Fail
T9	Check Remove Array with valid data after add node	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice: D ind: 0 indexOf: 0	Lists 5	As Expected	Pass
T10	Check Remove Array with invalid data after add node	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index	Choice: D ind=15 indexOf=15	Catch Error Handling java.lang.NullPointerException	As Expected	Fail

		E->Display size F->Display list Q->Exit first step				
T11	List is this 1 3 5 2 1	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice:F	Lists 1 3 5 2 1	As Expected	Pass
T12	List is this 1 3 2 5	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice: E	Size: 2	As Expected	Pass
T13	Check Add Array if capacity is over 1 3 5 2 1	A->Adding Node B->Adding array elements by node index	Choice: B ind: 0 data=8	Lists 1 3 5 2 1 8	As Expected	Pass

		C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step				
T14	Check Remove Array If one element 1 3 5 2 1 8	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice: D ind: 2 indexOf: 0	Lists 1 3 5 2 1	As Expected	Pass
T15	Check Quit then Double	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice:Q		As Expected	Pass
T16	Check Add	A->Adding	Choice: A	Lists	As Expected	Pass

	Node with Valid Data	Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	data: 1,2	1,2		
T17	Check Add Node with invalid Data	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice:A data: asdf	Java.util.InputMisMatch Exception	As Expected	Fail
T18	Check First Choice B or C or D	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display	Choice:D		As Expected	Fail

		size F->Display list Q->Exit first step				
T19	Check First Choice Size	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice:E	Size: 0	As Expected	Pass
T20	Check AddArray with valid data after add node	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice:B 0 5	Lists 1,20 5,00	As Expected	Pass
T21	Check AddArray with invalid data after add node	A->Adding Node B->Adding array elements by node index C->Delete	Choice:B 12 5	Java.lang.NullPointerException	As Expected	Fail

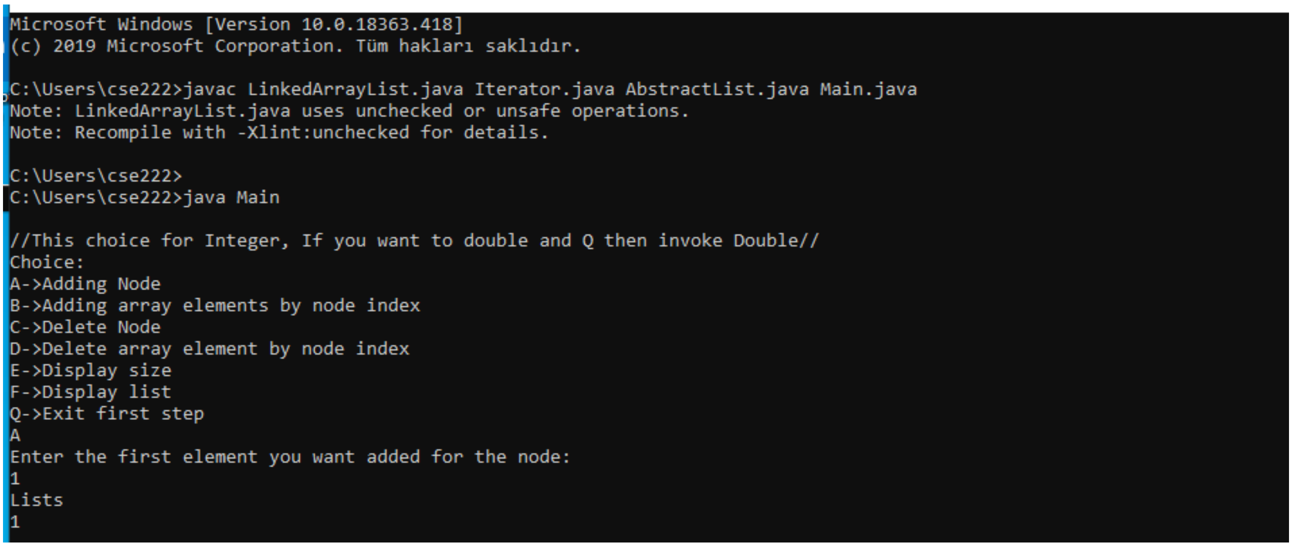
		Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step				
T22	Check AddArray with valid data after add node	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice: C 0	Lists 1,00 2,00	As Expected	Pass
T23	Check AddArray with invalid data after add node	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice: C 15	Lists 1,00 2,00 3,00 java.lang.IndexOutOfBoundsException: 15	As Expected	Fail
T24	Check Remove Array	A->Adding Node	Choice: D 0	Lists 2,00	As Expected	Pass

	with valid data after add node Lists 1,00 2,00	B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	0			
T25	Check Remove Array with invalid data after add node Lists 1,00 2,00	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice: D 15 15	Lists 1,00 2,00 java.lang.NullPointerException	As Expected	Fail
T26	List is this 2,00 1,30 1,00 1,33	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size	Choice: F	Lists 2,00 1,30 1,00 1,33	As Expected	Pass

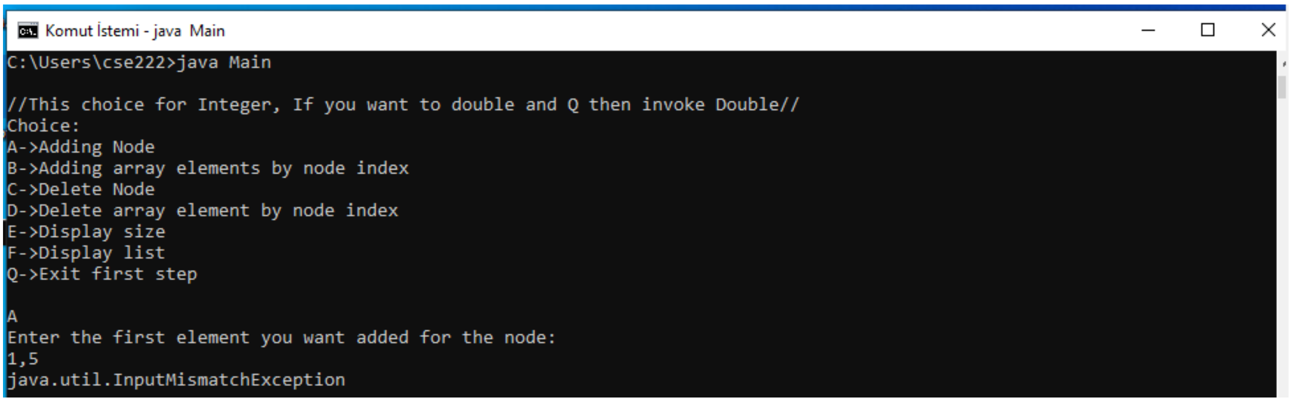
		F->Display list Q->Exit first step				
T27	List is this 2,00 1,30 1,00 1,33	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice: E	Size: 2	As Expected	Pass
T28	Check Add Array if capacity is over List is this 2,00 1,30 1,00 1,33 0,45 0,50 0,65	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice: B 2 8	List is this 2,00 1,30 1,00 1,33 0,45 0,50 0,65 8,00	As Expected	Pass
T29	Check Remove Array If one element List is this 2,00 1,30 1,00 1,33 0,45 0,50 0,65	A->Adding Node B->Adding array elements by node index C->Delete Node	Choice: D	List is this 2,00 1,30 1,00 1,33 0,45 0,50	As Expected	Pass

		D->Delete array element by node index E->Display size F->Display list Q->Exit first step				
T30	Check Quit	A->Adding Node B->Adding array elements by node index C->Delete Node D->Delete array element by node index E->Display size F->Display list Q->Exit first step	Choice:Q	Exit..	As Expected	Pass

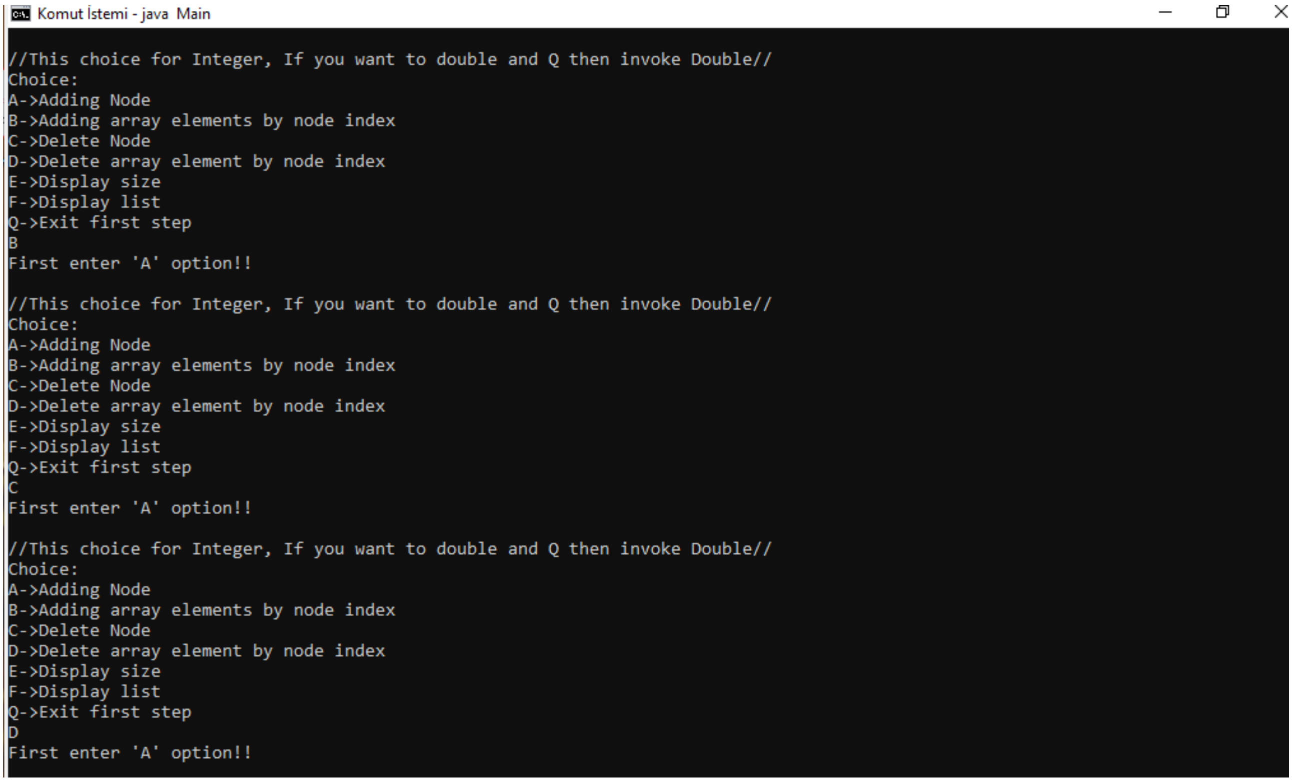
4. RUNNING AND RESULTS



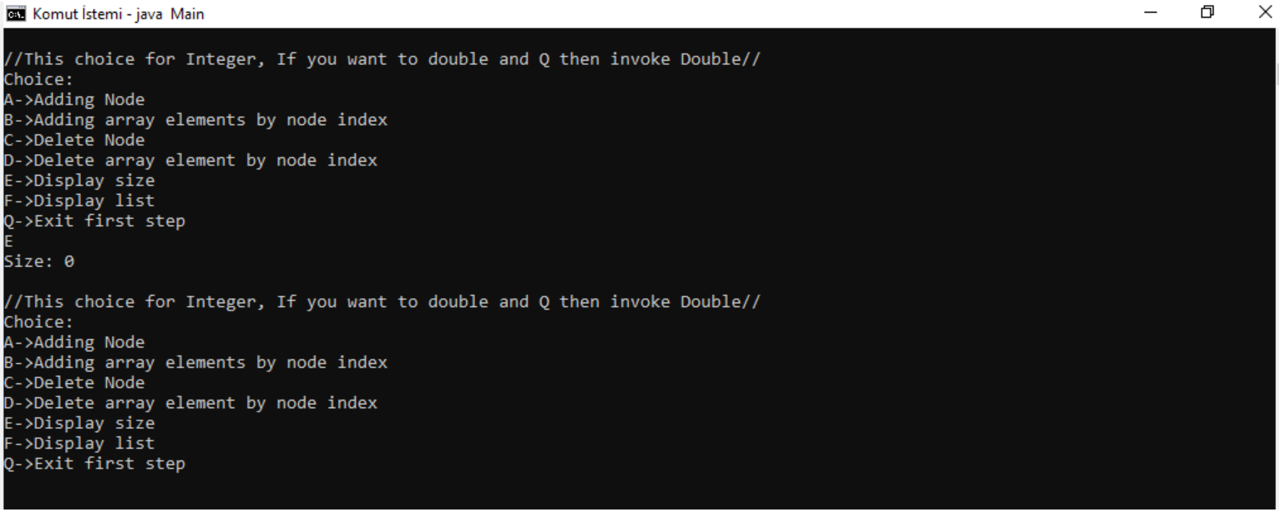
```
Microsoft Windows [Version 10.0.18363.418]
(c) 2019 Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\cse222>javac LinkedArrayList.java Iterator.java AbstractList.java Main.java
Note: LinkedArrayList.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

C:\Users\cse222>
C:\Users\cse222>java Main

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
A
Enter the first element you want added for the node:
1
Lists
1
```



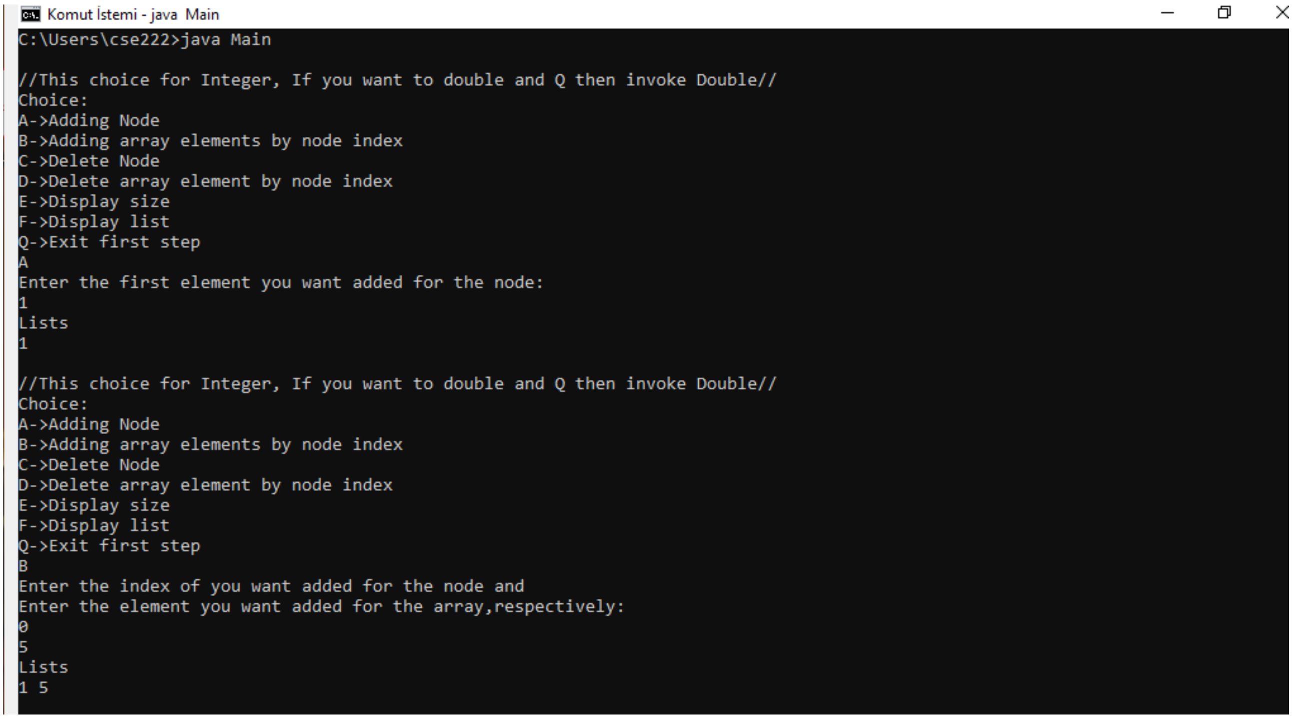
```
C:\Users\cse222>java Main

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step

A
Enter the first element you want added for the node:
1,5
java.util.InputMismatchException
```



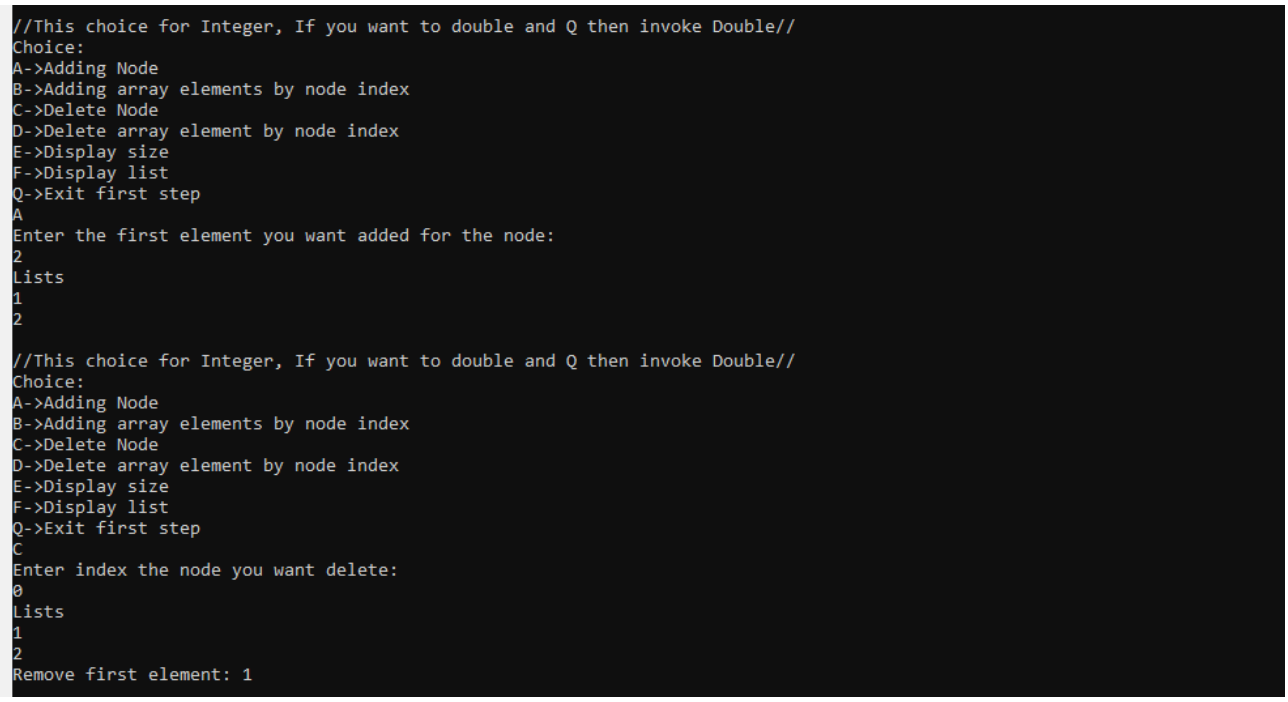
```
//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
B
First enter 'A' option!!

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
C
First enter 'A' option!!

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
D
First enter 'A' option!!
```



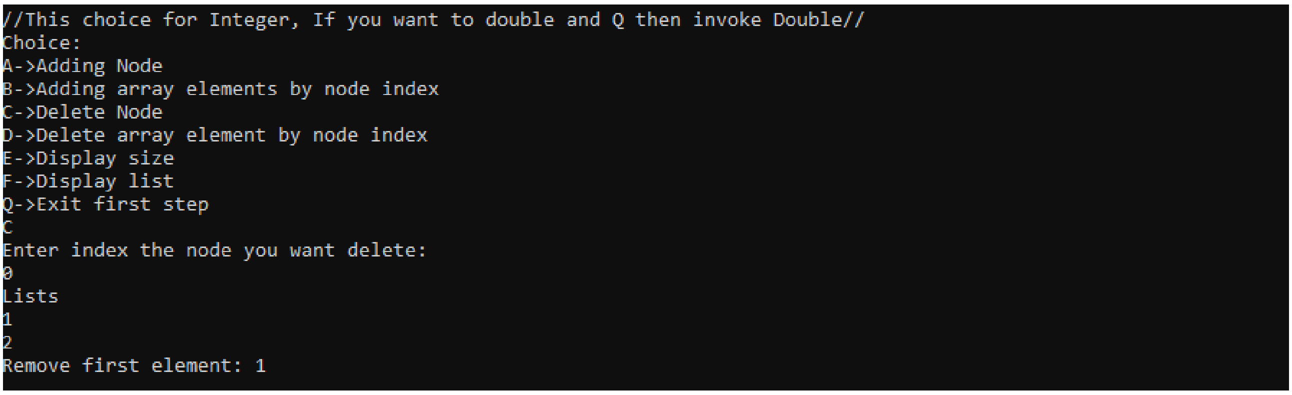
```
//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
E
Size: 0

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
```



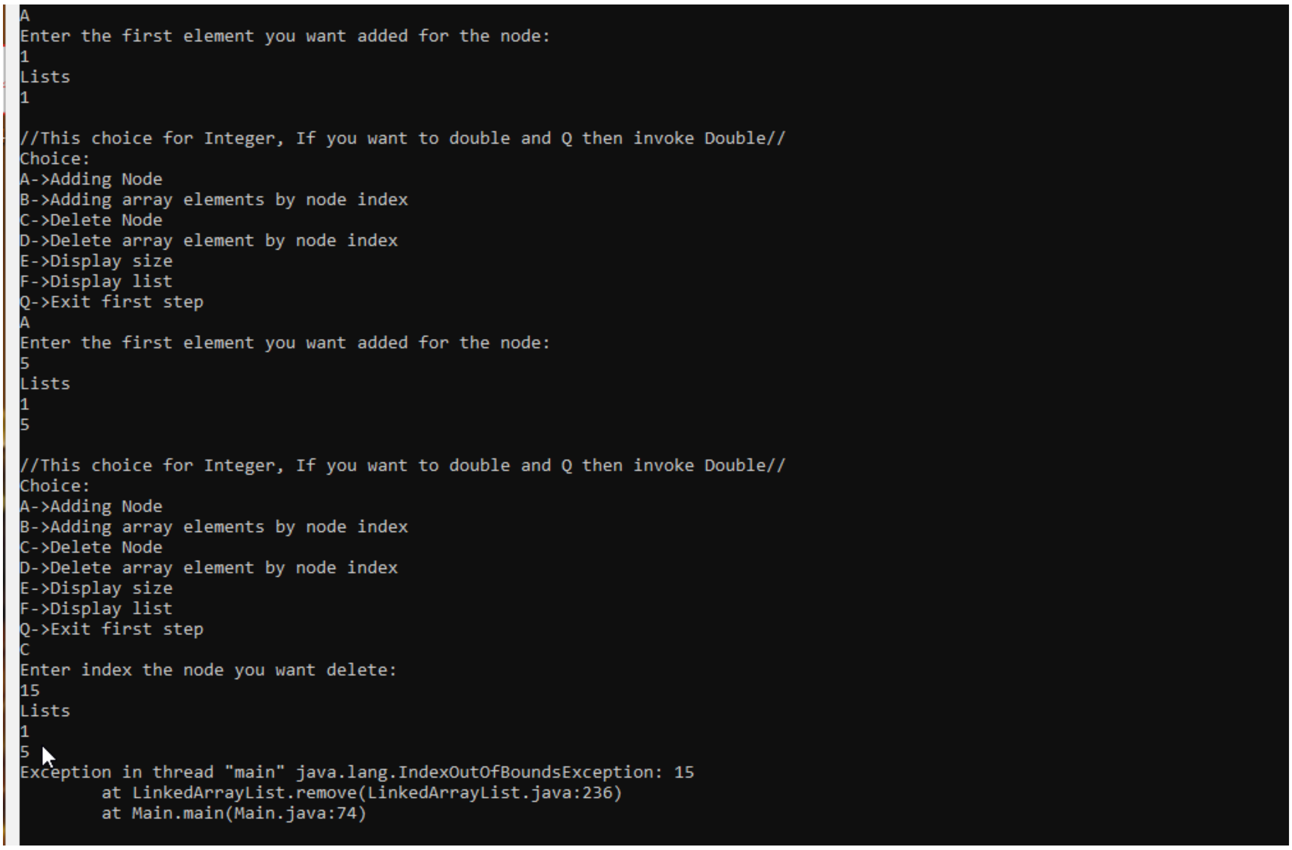
```
C:\Users\cse222>java Main

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
A
Enter the first element you want added for the node:
1
Lists
1

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
B
Enter the index of you want added for the node and
Enter the element you want added for the array,respectively:
0
5
Lists
1 5
```



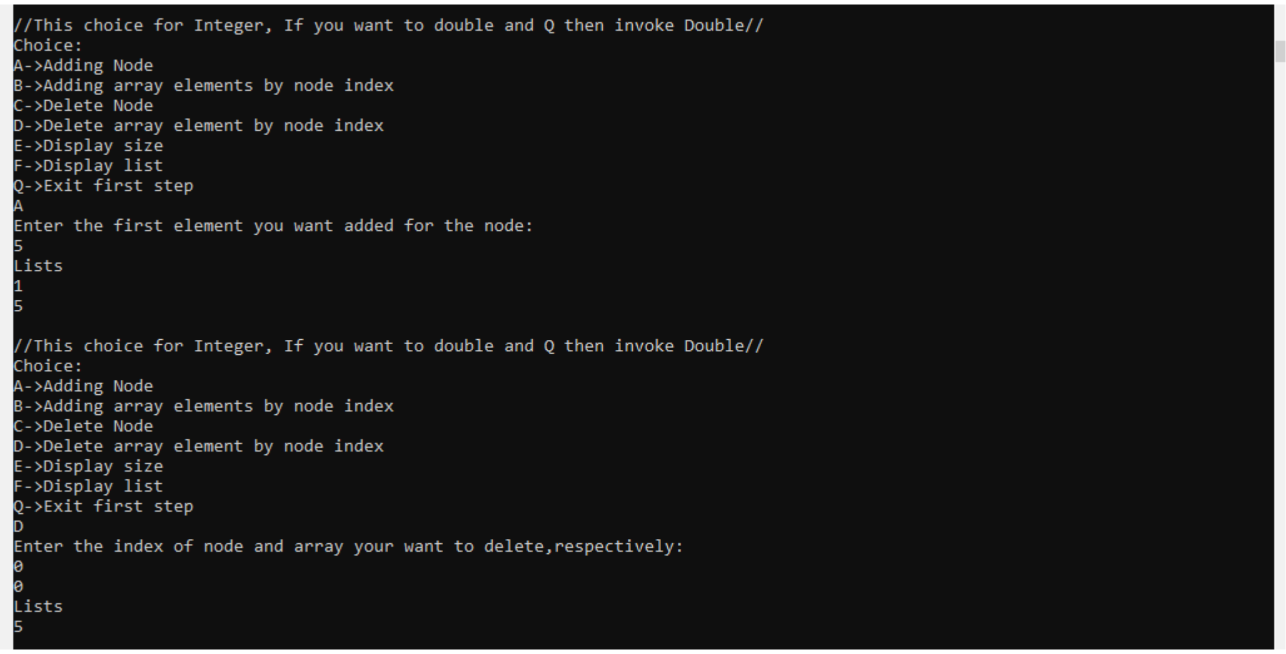
```
//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
A
Enter the first element you want added for the node:
2
Lists
1
2

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
C
Enter index the node you want delete:
0
Lists
1
2
Remove first element: 1
```



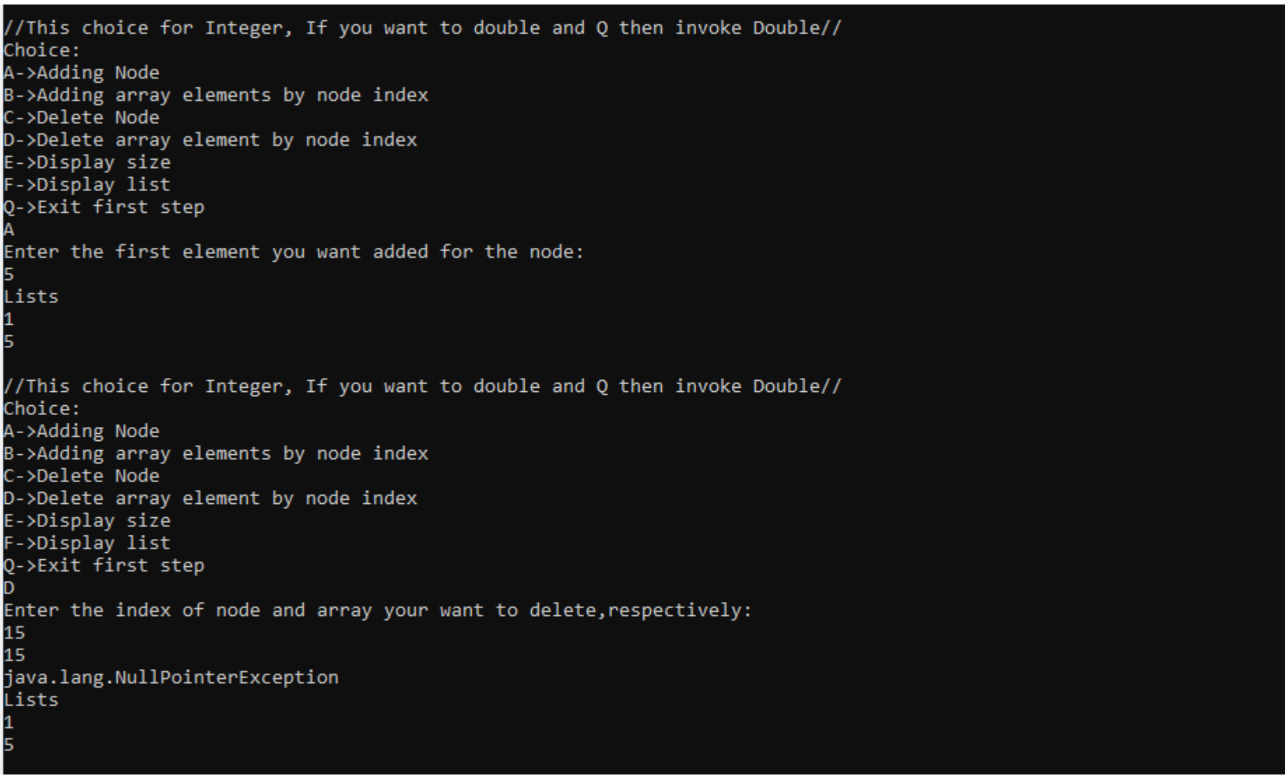
```
//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
C
Enter index the node you want delete:
0
Lists
1
2
Remove first element: 1
```



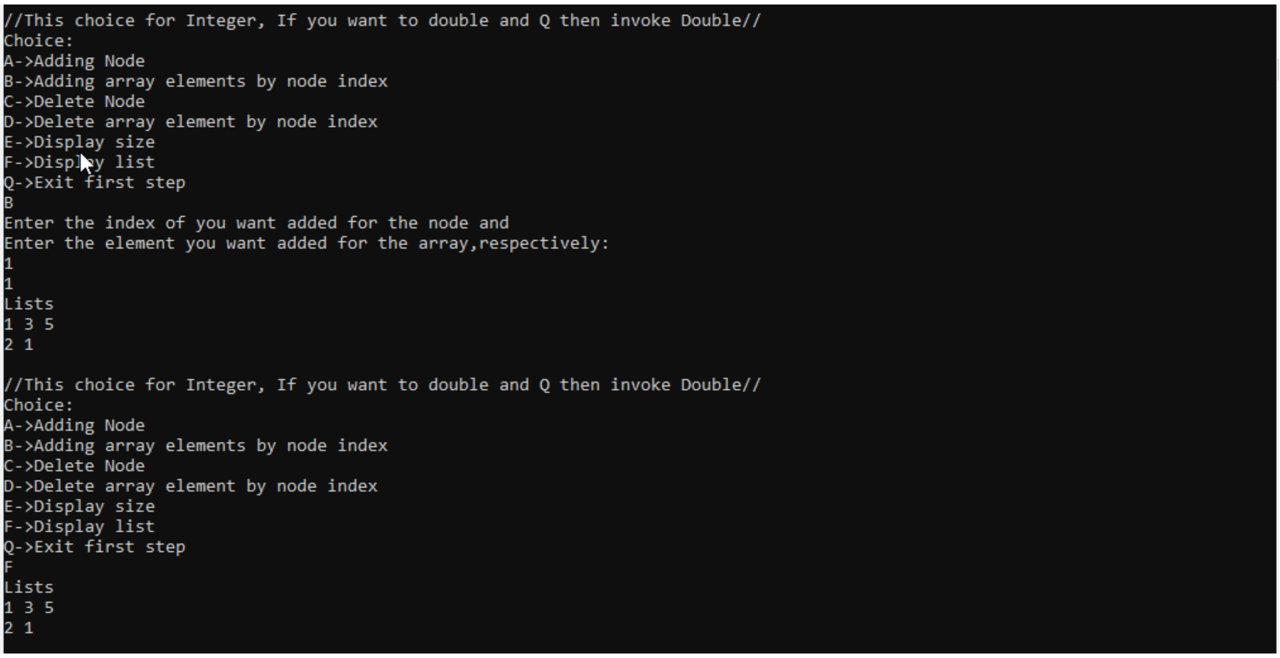
```
A
Enter the first element you want added for the node:
1
Lists
1

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
A
Enter the first element you want added for the node:
5
Lists
1
5

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
C
Enter index the node you want delete:
15
Lists
1
5
Exception in thread "main" java.lang.IndexOutOfBoundsException: 15
        at LinkedArrayList.remove(LinkedArrayList.java:236)
        at Main.main(Main.java:74)
```



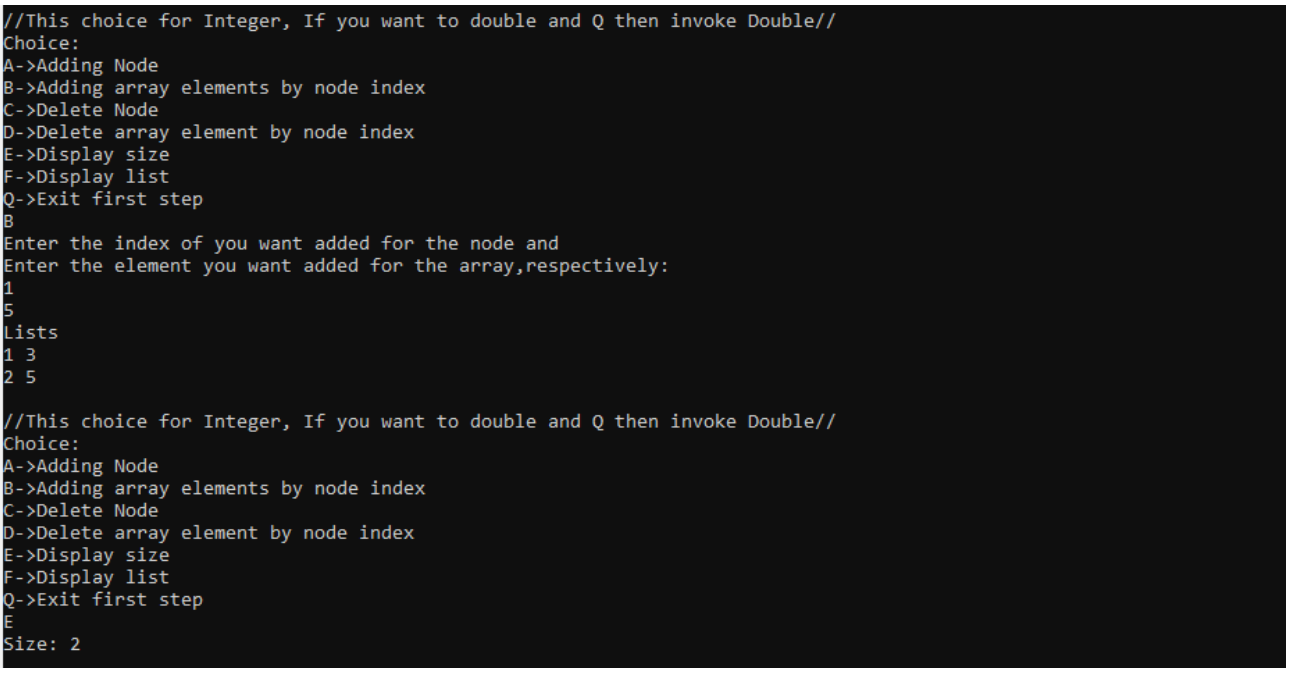
```
//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
A
Enter the first element you want added for the node:
5
Lists
1
5

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
D
Enter the index of node and array your want to delete,respectively:
0
0
Lists
5
```



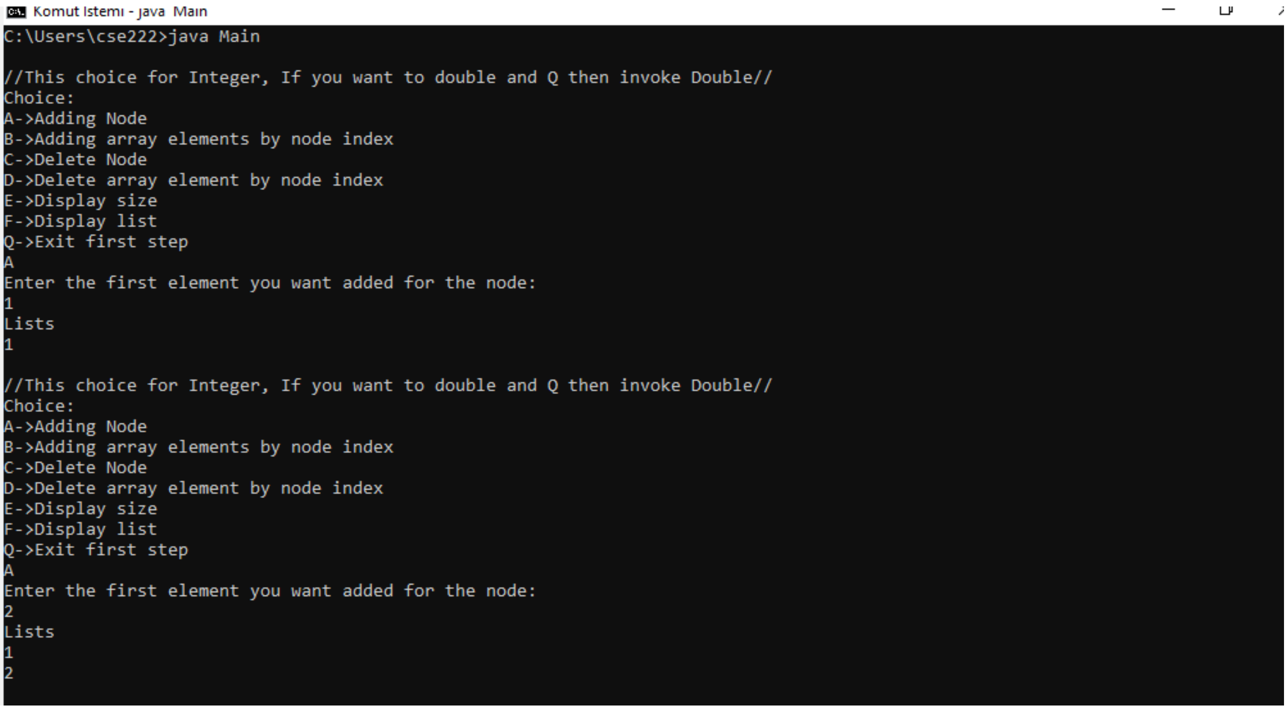
```
//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
A
Enter the first element you want added for the node:
5
Lists
1
5

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
D
Enter the index of node and array your want to delete,respectively:
15
15
java.lang.NullPointerException
Lists
1
5
```



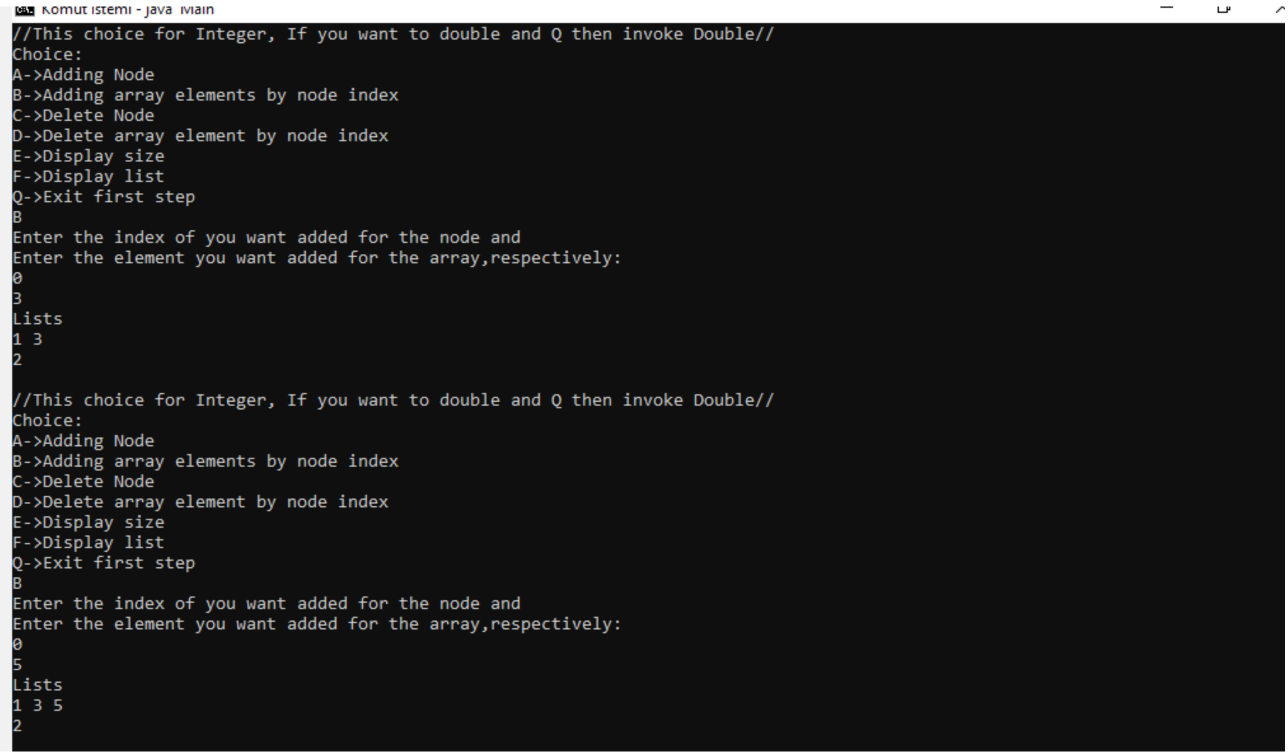
```
//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
B
Enter the index of you want added for the node and
Enter the element you want added for the array,respectively:
1
1
Lists
1 3 5
2 1

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
F
Lists
1 3 5
2 1
```



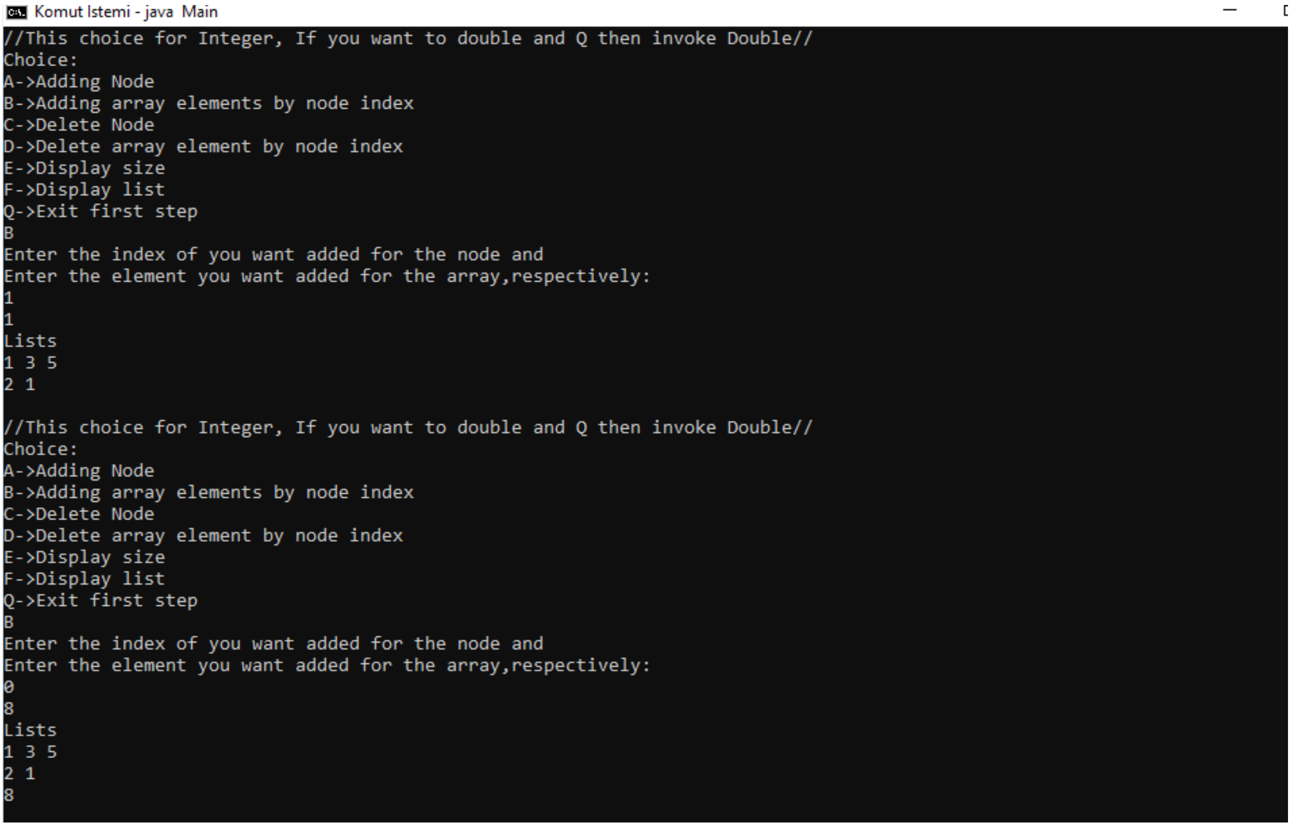
```
//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
B
Enter the index of you want added for the node and
Enter the element you want added for the array,respectively:
1
5
Lists
1 3
2 5

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
E
Size: 2
```



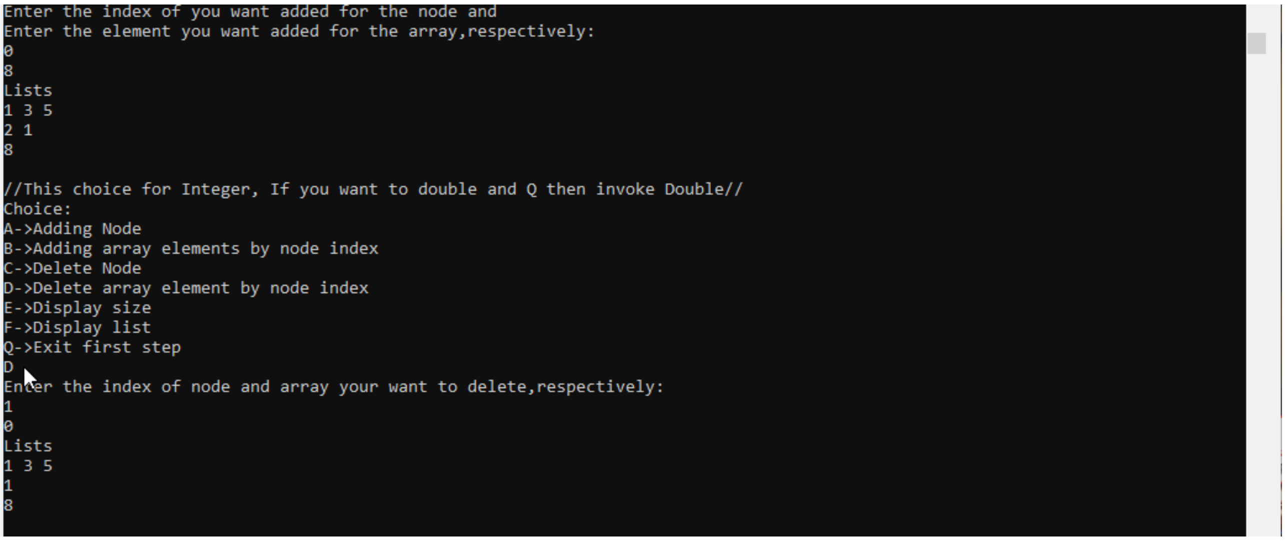
```
C:\Users\cse222>java Main

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
A
Enter the first element you want added for the node:
1
Lists
1

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
A
Enter the first element you want added for the node:
2
Lists
1
2
```



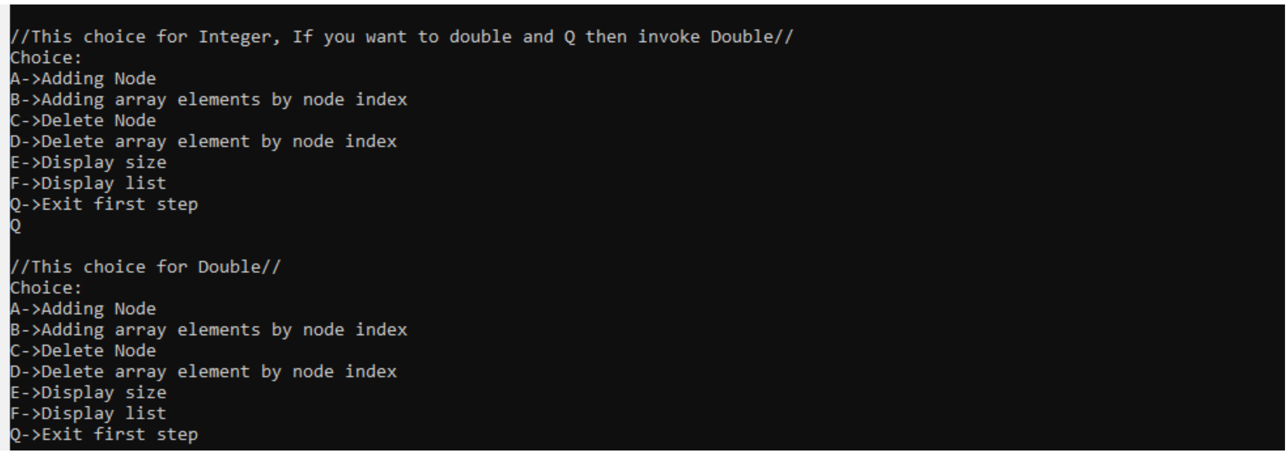
```
//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
B
Enter the index of you want added for the node and
Enter the element you want added for the array,respectively:
0
3
Lists
1 3
2

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
B
Enter the index of you want added for the node and
Enter the element you want added for the array,respectively:
0
5
Lists
1 3 5
2
```



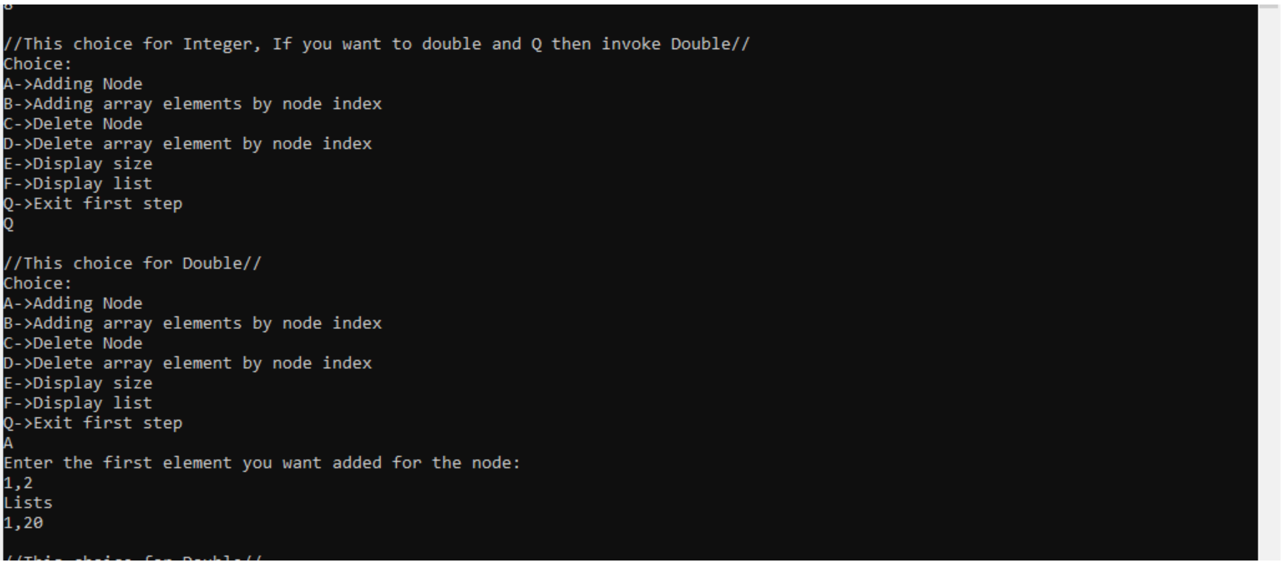
```
//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
B
Enter the index of you want added for the node and
Enter the element you want added for the array,respectively:
1
1
Lists
1 3 5
2 1

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
B
Enter the index of you want added for the node and
Enter the element you want added for the array,respectively:
0
8
Lists
1 3 5
2 1
8
```



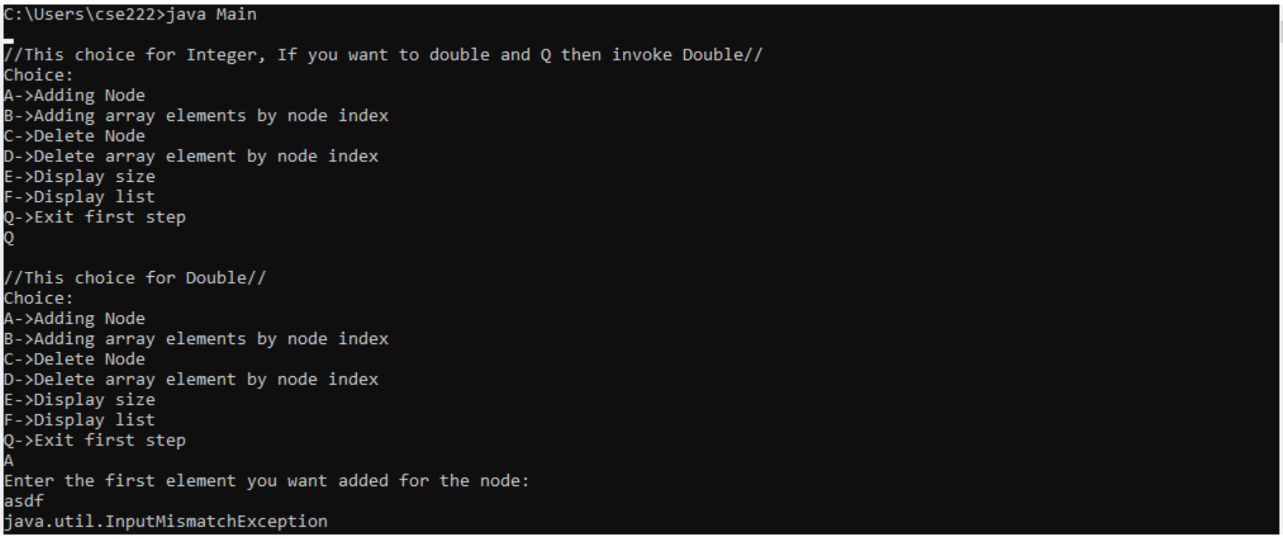
```
Enter the index of you want added for the node and
Enter the element you want added for the array,respectively:
0
8
Lists
1 3 5
2 1
8

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
D
Enter the index of node and array your want to delete,respectively:
1
0
Lists
1 3 5
1
8
```



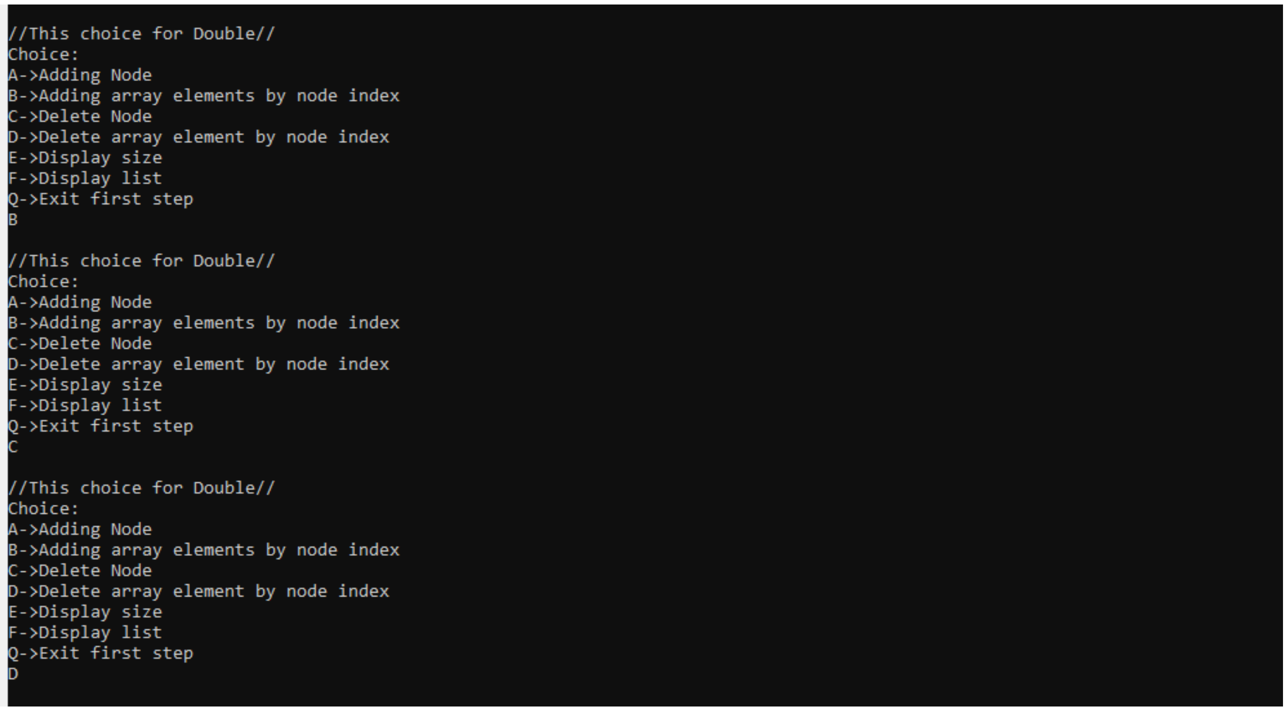
```
//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
Q

//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
```



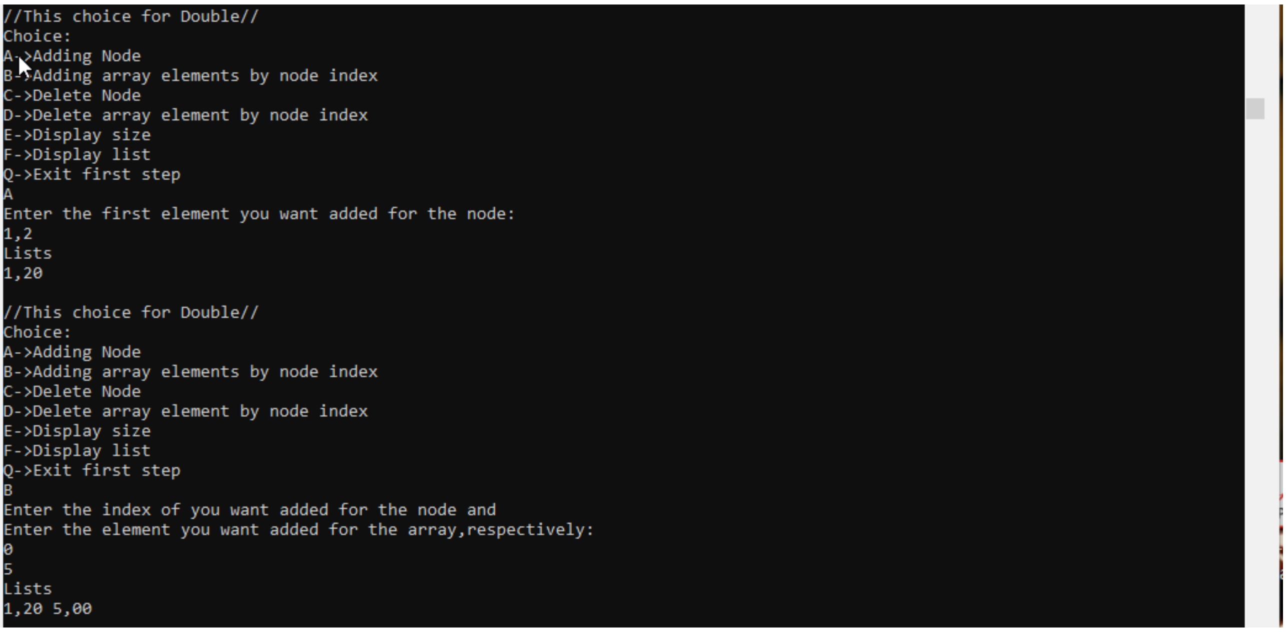
```
//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
Q

//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
A
Enter the first element you want added for the node:
1,2
Lists
1,20

//This choice for Double//
```



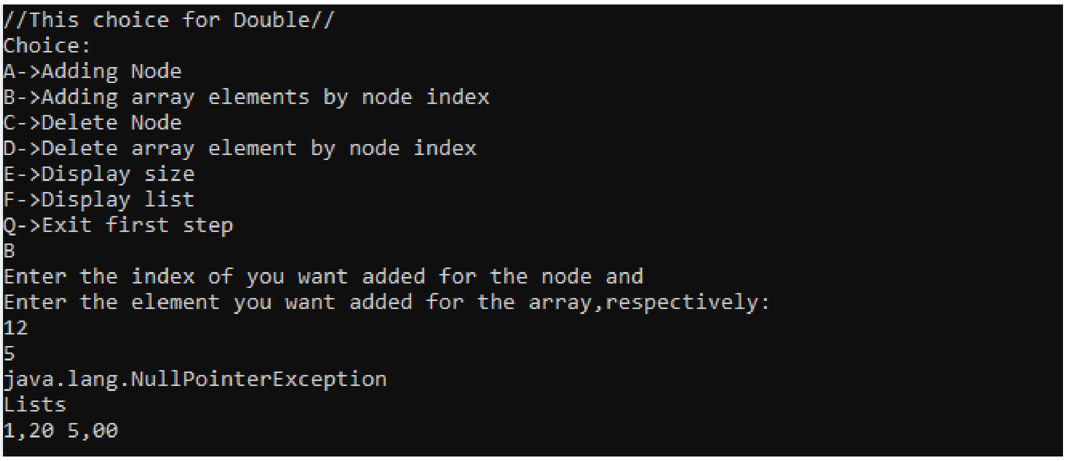
```
C:\Users\cse222>java Main

//This choice for Integer, If you want to double and Q then invoke Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
Q

//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
A
Enter the first element you want added for the node:
asdf
java.util.InputMismatchException
```



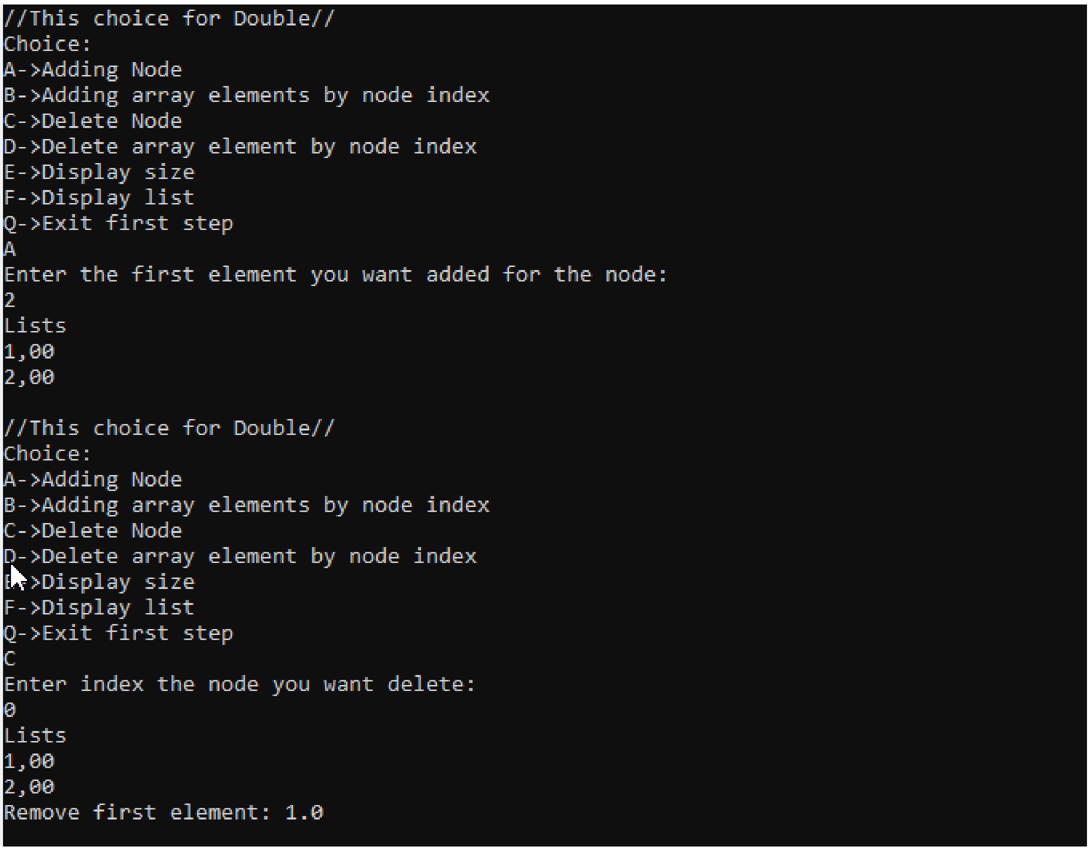
```
//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
B

//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
C

//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
D
```



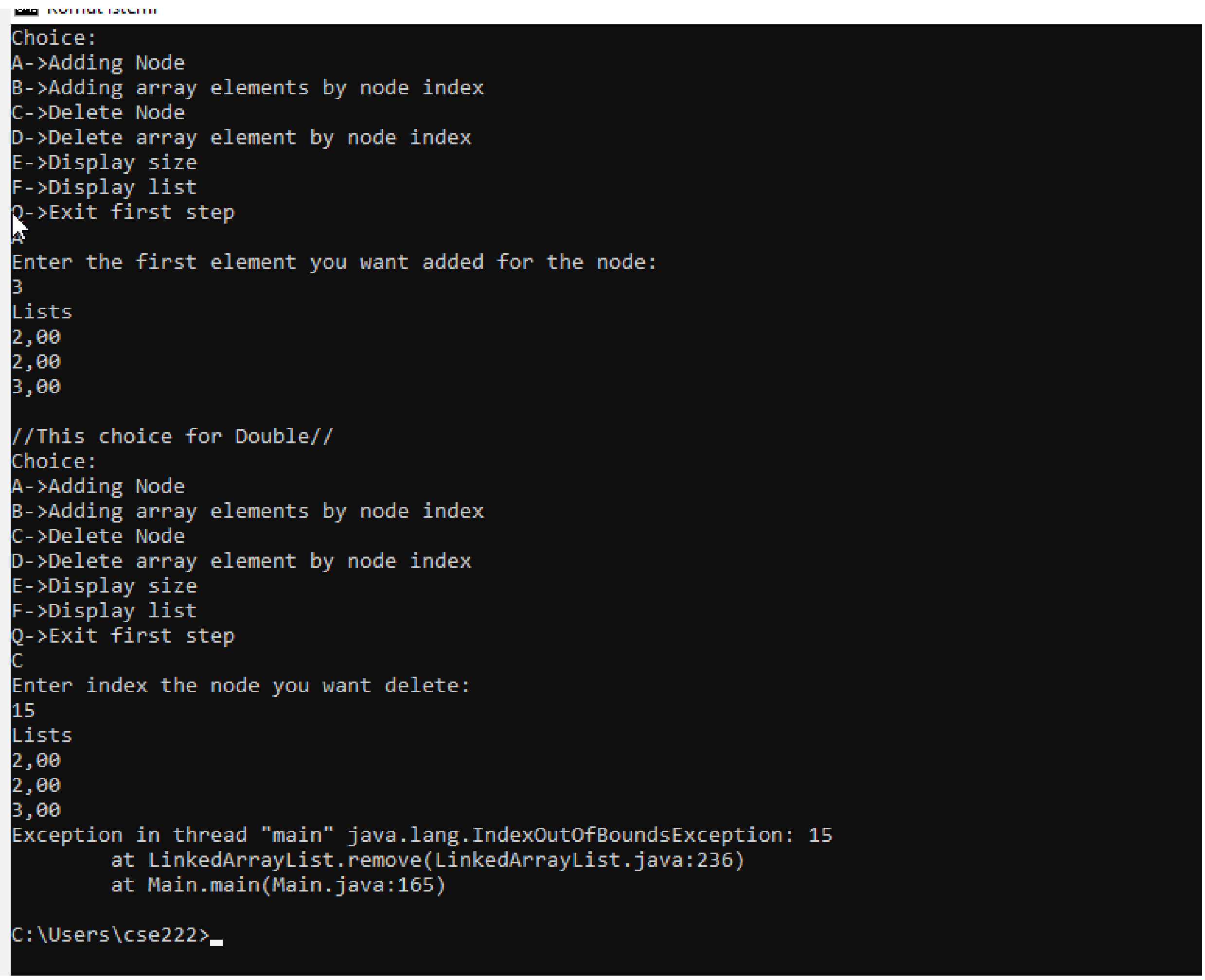
```
//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
A
Enter the first element you want added for the node:
1,2
Lists
1,20

//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
B
Enter the index of you want added for the node and
Enter the element you want added for the array,respectively:
0
5
Lists
1,20 5,00
```



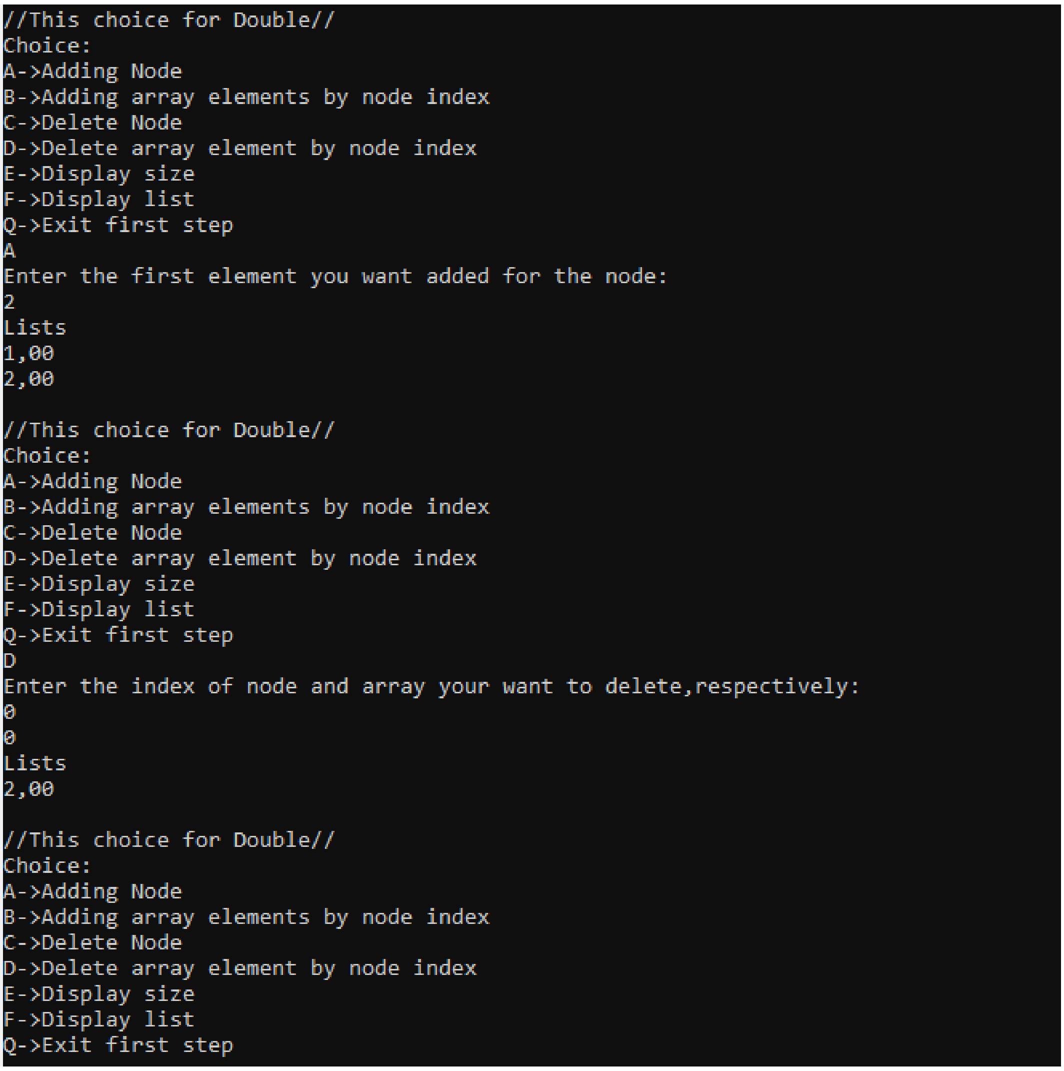
```
//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
B
Enter the index of you want added for the node and
Enter the element you want added for the array,respectively:
12
5
java.lang.NullPointerException
Lists
1,20 5,00
```



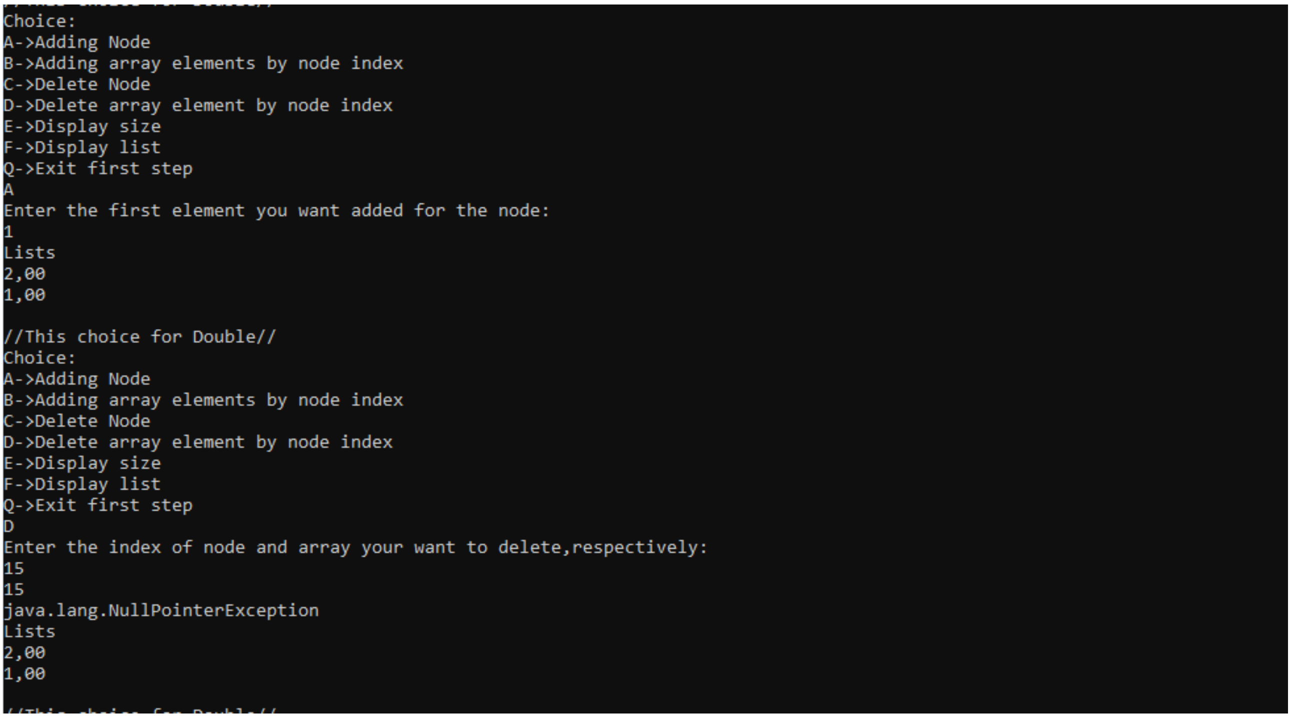
```
//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
A
Enter the first element you want added for the node:
2
Lists
1,00
2,00

//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
C
Enter index the node you want delete:
0
Lists
1,00
2,00
Remove first element: 1.0
```



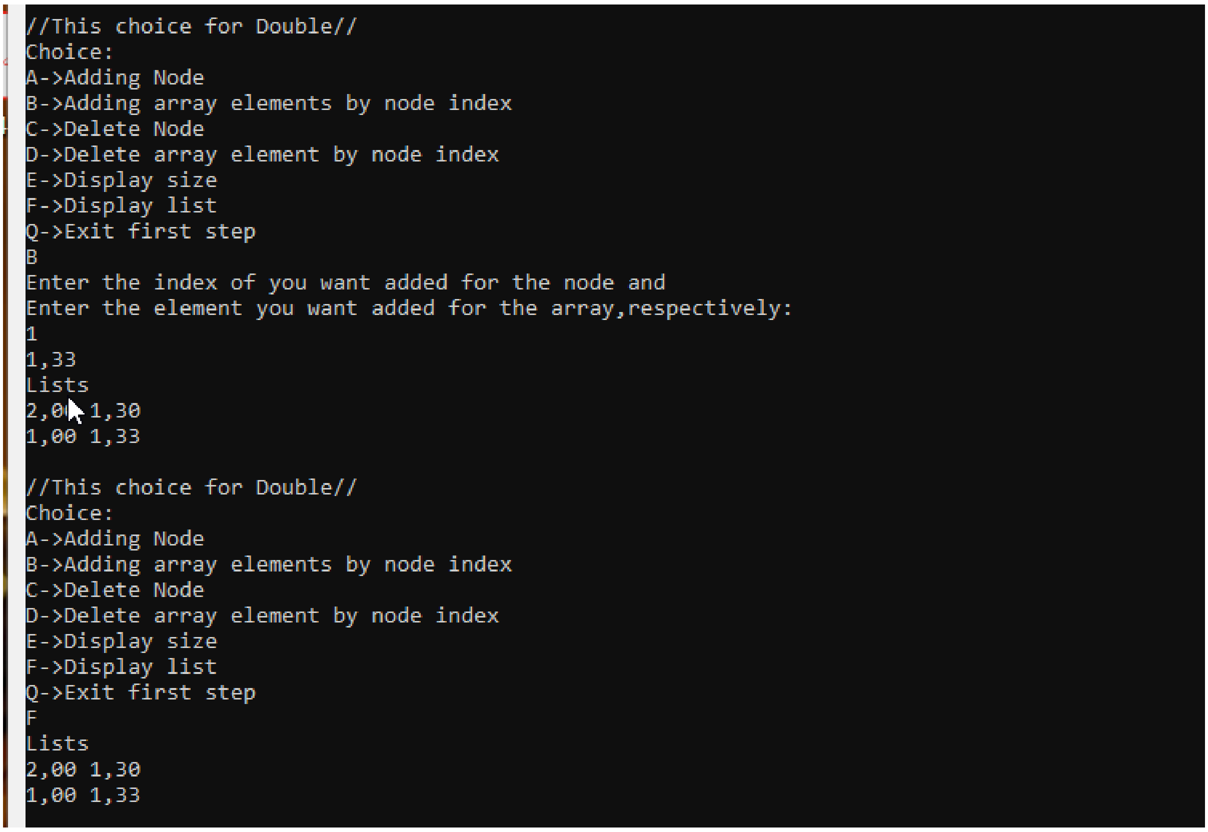
```
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
A
Enter the first element you want added for the node:
3
Lists
2,00
2,00
3,00

//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
C
Enter index the node you want delete:
15
Lists
2,00
2,00
3,00
Exception in thread "main" java.lang.IndexOutOfBoundsException: 15
        at LinkedArrayList.remove(LinkedArrayList.java:236)
        at Main.main(Main.java:165)

C:\Users\cse222>
```



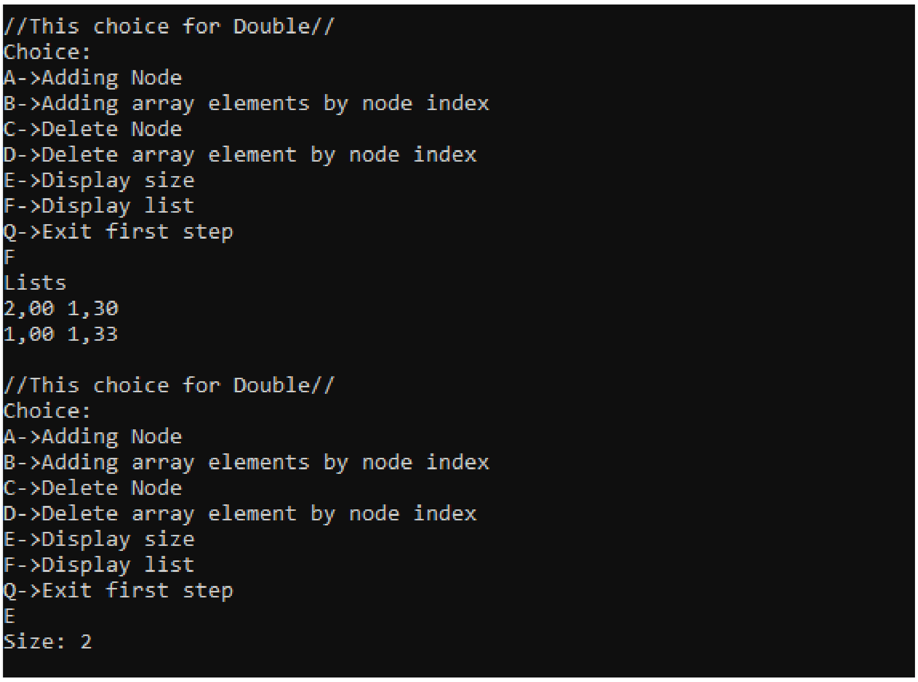
```
//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
A
Enter the first element you want added for the node:
2
Lists
1,00
2,00

//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
D
Enter the index of node and array your want to delete,respectively:
0
0
Lists
2,00

//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
```



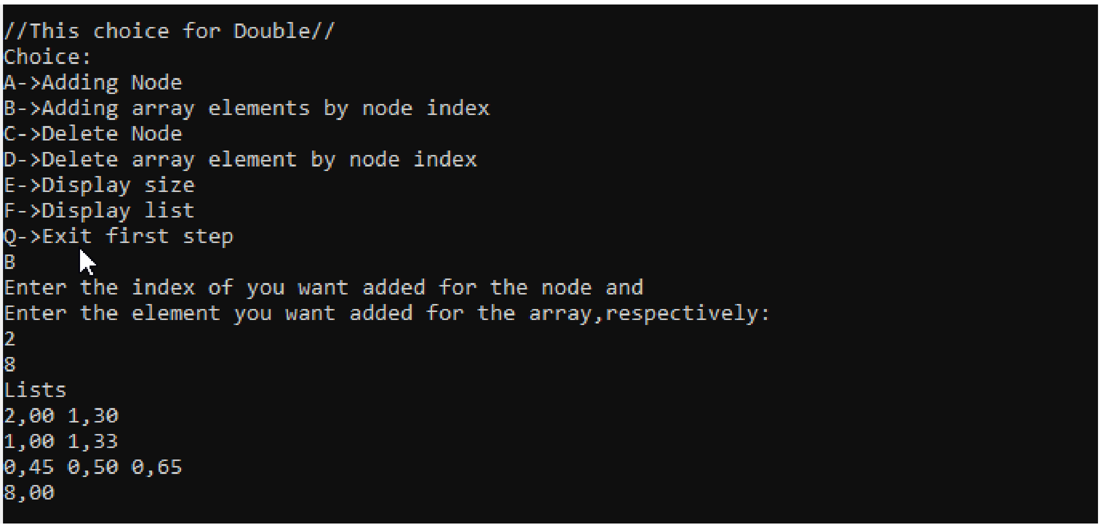
```
//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
A
Enter the first element you want added for the node:
1
Lists
2,00
1,00

//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
D
Enter the index of node and array your want to delete,respectively:
15
15
java.lang.NullPointerException
Lists
2,00
1,00

//This choice for Double//
```



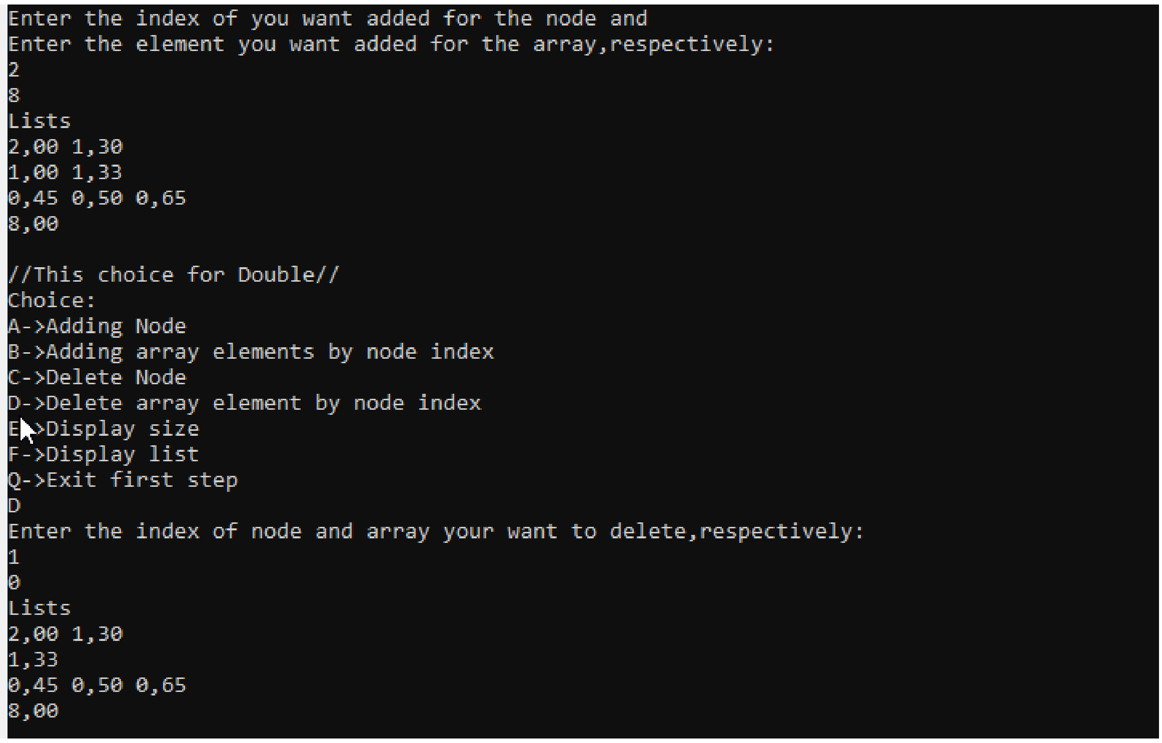
```
//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
B
Enter the index of you want added for the node and
Enter the element you want added for the array,respectively:
1
1,33
Lists
2,00 1,30
1,00 1,33

//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
F
Lists
2,00 1,30
1,00 1,33
```



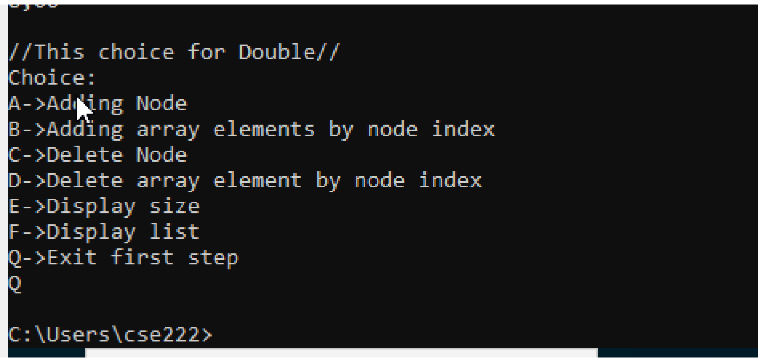
```
//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
F
Lists
2,00 1,30
1,00 1,33

//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
E
Size: 2
```



```
//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
B
Enter the index of you want added for the node and
Enter the element you want added for the array,respectively:
2
8
Lists
2,00 1,30
1,00 1,33
0,45 0,50 0,65
8,00
```



```
Enter the index of you want added for the node and
Enter the element you want added for the array,respectively:
2
8
Lists
2,00 1,30
1,00 1,33
0,45 0,50 0,65
8,00

//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
D
Enter the index of node and array your want to delete,respectively:
1
0
Lists
2,00 1,30
1,33
0,45 0,50 0,65
8,00
```



```
//This choice for Double//
Choice:
A->Adding Node
B->Adding array elements by node index
C->Delete Node
D->Delete array element by node index
E->Display size
F->Display list
Q->Exit first step
Q

C:\Users\cse222>
```

#Question 2#

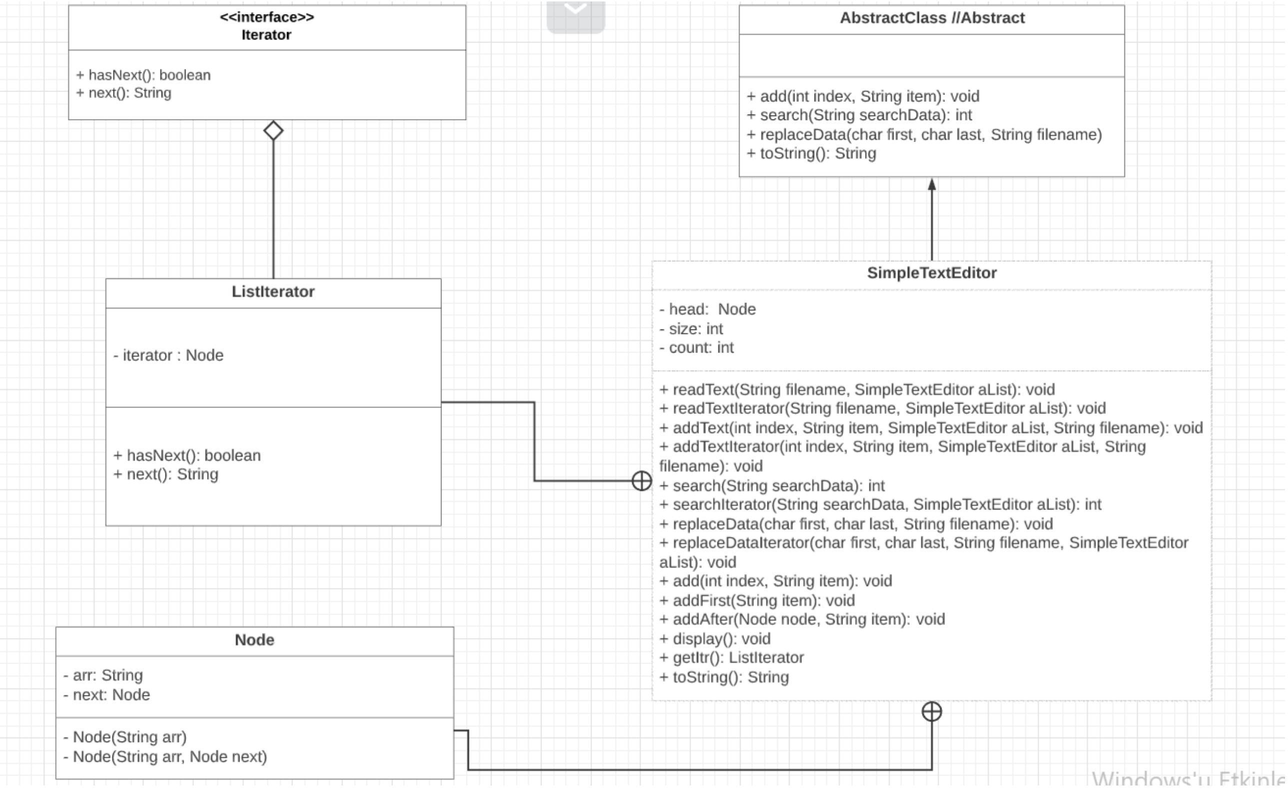
1.BIG O NOTATION

List is an ArrayList and iterator not used add method-> ArrayList occupies a certain capacity and we use the loop as we can add elements to the index we want. The loop advances linearly. So, asymptotic notation is O(n).
List is an ArrayList and iterator used add method-> In the case of the iterator, only the following changes, the iterator next () method runs constant time, so asymptotic notation O(n)
List is an ArrayList and iterator not used read method-> Since we throw ArrayList after reading it becomes O (n).
List is an ArrayList and iterator used read method-> The iterator next () method runs constant time, O(n).
List is an ArrayList and iterator not used search method-> We search for elements in the ArrayList and we reach the word through a loop.So the worst case is O (n). Asymptotic notation is O(n).
List is an ArrayList and iterator used search method-> Asymptotic notation is O(n).
List is an ArrayList and iterator not used replace method-> To change the element, it is necessary to access the element.ArrayList access method time complexity is O(1).Progress happens at constat time. Asymptotic notation is O(1).
List is an ArrayList and iterator not used replace method-> There is no change when used with Iterator.Progress happens at constat time. Asymptotic notation is O(1).
List is an LinkedList and iterator not used add method-> There is no loop in the node. Since there is an addition to the node from the beginning or the end, there is a progress in a constant time.So, asymptotic notation is O(1).
List is an LinkedList and iterator used add method-> When used with iterator, there is no change. The next () method takes place in constant time. So, asymptotic notation is O(1).
List is an LinkedList and iterator not used read method-> There is no loop. This method is progressing in constant time O(1). Asypmtotic notation is O(1).
List is an LinkedList and iterator used read method-> When used with iterator, there is no change. The next () method takes place in constant time. So, asymptotic notation is O(1).
List is an LinkedList and iterator not used search method-> There is an element search within a node and we go through the loop in the word and look for it. If he finds it in the first element, the best case is O (1). But if it doesn't find a probe or at all, it becomes O (n).

List is an LinkedList and iterator used search method-> There is an element search in a node and it proceeds with the iterator. The next () method of the iterator is constant time.So, asymptotic notation is O(1).
List is an LinkedList and iterator not used replace method-> To change an element, we must first go through the loop and find it. So, asymptotic notation is O(n).
List is an LinkedList and iterator used replace method->Asymptotic notation is O(n).

Experimental performance
Double Linked List add method-> If there is one node, best case O(1) but if there is more than one worst case is O(n).
Double Linked List search method-> There is a linear increase, as it is necessary to find the data by looping through it. So, asymptotic notation is O(n).
Double Linked List replace method-> Asymptotic notation is O(n).

2. CLASS DIAGRAMS



<<interface>>
Iterator
+ hasNext(): boolean
+ next(): String
AbstractClass //Abstract
+ add(int index, String item): void
+ search(String searchData): int
+ replaceData(char first, char last, String filename)
+ toString(): String
ListIterator
- iterator : Node
+ hasNext(): boolean
+ next(): String
SimpleTextEditor
- head: Node
- size: int
- count: int
+ readText(String filename, SimpleTextEditor aList): void
+ readTextIterator(String filename, SimpleTextEditor aList): void
+ addText(int index, String item, SimpleTextEditor aList, String filename): void
+ addTextIterator(int index, String item, SimpleTextEditor aList, String filename): void
+ search(String searchData): int
+ searchIterator(String searchData, SimpleTextEditor aList): int
+ replaceData(char first, char last, String filename): void
+ replaceDataIterator(char first, char last, String filename, SimpleTextEditor aList): void
+ add(int index, String item): void
+ addFirst(String item): void
+ addAfter(Node node, String item): void
+ display(): void
+ getItr(): ListIterator
+ toString(): String
Node
- arr: String
- next: Node
- Node(String arr)
- Node(String arr, Node next)

3. PROBLEM SOLUTION APPROACH

Our problem with this question is the relationship between text and list. We take data from the text and create nodes in the list and throw the data into the nodes. We write the data we want according to the index. We find the element you want and return index. We can enter characters and write another character instead of that character. We do this both with iterator and without iterators. While solving the problem, we have a 2-stage menu in the testing phase. One of the two files (dosya.txt, dosya2.txt) I have chosen is selected. Different sizes in two files. We have a total of 9 options in the menu. A->Read Text and Construct,Iterator not used, B->Add text by index, iterator not used, C->Search characters then return index, iterator not used, D->Replace character by character, iterator not used, E->Read Text and Construct,Iterator not used F->Add text by index, iterator not used, G->Search characters then return index, iterator not use,H->Replace character by character, iterato, Q->Quit . We also measure the time in each option of this menu.

We read the files from the text in the Read Text and add them to the nodes of the list. In add text, we enter indexi and data and add it to the node, and then we write the node in text. The file is personally moving to the new line in every 3 words. In search element, return index if the element we want to find, otherwise return -1. Replace element enters and replaces the character we want to change and will change.These 4 methods are iteratorized in the remaining 4 options.

4. TEST CASES

Test Case ID	Test Scenario	Test Steps	Test Data	Excepted Results	Actual Results	Pass/Fail
T1	Check Choice Read Text without iterator	A->Read Text and Construct,Iterator not used B->Add text by index, iterator not used C->Search characters then return index, iterator not used D->Replace character by character, iterator not used E->Read Text and Construct,Iterator not used	Choice: A	x1 Xx1 ABC 123 x2 DEF 1903 xx3 TRY 13 x99 Aaa	As Expected	Pass

		F->Add text by index, iterator not used G->Search characters then return index, iterator not used H->Replace character by character, iterator not used Q->Quit				
T2	Check First Choice Add Text without iterator	A->Read Text and Construct,Iterator not used B->Add text by index, iterator not used C->Search characters then return index, iterator not used D->Replace character by character, iterator not used E->Read Text and Construct,Iterator not used F->Add text by index, iterator not used G->Search characters then return index, iterator not used H->Replace character by character, iterator not used Q->Quit	Choice:B	First Enter ‘A’ or ‘E’ option	As Expected	Fail
T3	Check Choice Add Text After Read Text	A->Read Text and Construct,Iterator not used B->Add text by index, iterator not used C->Search characters then return index, iterator not used D->Replace character by character, iterator not used E->Read Text and Construct,Iterator not used F->Add text by index, iterator not used G->Search characters	Choice: B 3 333	x1 Xx1 ABC 333 123 x2 DEF 1903 xx3 TRY 13 x99 Aaa	As Expected	Pass

		then return index, iterator not used H->Replace character by character, iterator not used Q->Quit				
T4	Check Choice Search After Read Text valid data	A->Read Text and Construct,Iterator not used B->Add text by index, iterator not used C->Search characters then return index, iterator not used D->Replace character by character, iterator not used E->Read Text and Construct,Iterator not used F->Add text by index, iterator not used G->Search characters then return index, iterator not used H->Replace character by character, iterator not used Q->Quit	Choice: B Enter of search string data x1	index=0 Search data without iterator time in miliseconds :6241	As Expected	Pass
T5	Check Choice Search After Read Text invalid data	A->Read Text and Construct,Iterator not used B->Add text by index, iterator not used C->Search characters then return index, iterator not used D->Replace character by character, iterator not used E->Read Text and Construct,Iterator not used F->Add text by index, iterator not used G->Search characters then return index, iterator not used H->Replace character by character, iterator not used	Choice: C Enter of search string data: asd	Not Found this element Search data without iterator time im miliseconds :3286	As Expected	Pass

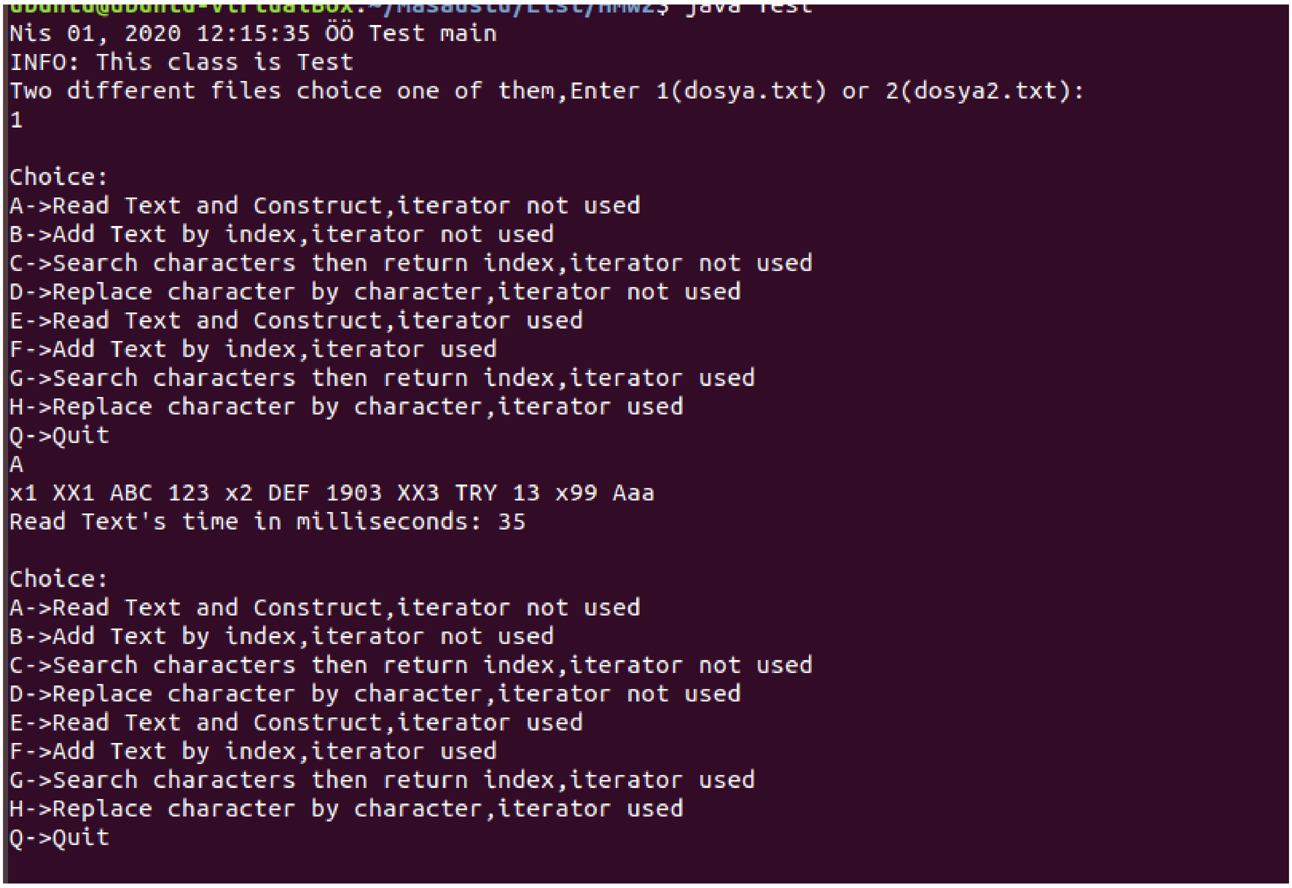
		Q->Quit				
T6	Check Choice Replacement data After Read Text valid data	A->Read Text and Construct,Iterator not used B->Add text by index, iterator not used C->Search characters then return index, iterator not used D->Replace character by character, iterator not used E->Read Text and Construct,Iterator not used F->Add text by index, iterator not used G->Search characters then return index, iterator not used H->Replace character by character, iterator not used Q->Quit	Choice: D Enter of element you want to change and replacement elements, repectively	Replace data without iterator time in miliseconds : 8146	As Expected	Pass
T7	Check Choice Read Text after Read Text	A->Read Text and Construct,Iterator not used B->Add text by index, iterator not used C->Search characters then return index, iterator not used D->Replace character by character, iterator not used E->Read Text and Construct,Iterator not used F->Add text by index, iterator not used G->Search characters then return index, iterator not used H->Replace character by character, iterator not used Q->Quit	Choice:A x1 Xx1 ABC 333 123 x2 DEF 1903 xx3 TRY 13 x99 Aaa Read Text’s time in miliseconds : 38 Choice: A	x1 Xx1 ABC 333 123 x2 DEF 1903 xx3 TRY 13 x99 Aaa Read Text’s time in miliseconds : 9	As Expected	Pass
T8	Check Choice Search Iterator for read iterator	A->Read Text and Construct,Iterator not used B->Add text by index,	Choice:F Enter of integer index for	x1 ADD Xx1 ABC 333 123 x2 DEF 1903	As Expected	Pass

	text valid data	iterator not used C->Search characters then return index, iterator not used D->Replace character by character, iterator not used E->Read Text and Construct,Iterator not used F->Add text by index, iterator not used G->Search characters then return index, iterator not used H->Replace character by character, iterator not used Q->Quit	adding text and enter of string data: 1 ADD	xx3 TRY 13 x99 Aaa Search Data with iterator in miliseconds :1594		
T9	Check Choice Search for read iterator text valid data	A->Read Text and Construct,Iterator not used B->Add text by index, iterator not used C->Search characters then return index, iterator not used D->Replace character by character, iterator not used E->Read Text and Construct,Iterator not used F->Add text by index, iterator not used G->Search characters then return index, iterator not used H->Replace character by character, iterator not used Q->Quit	Choice:F Enter of integer index for adding text and enter ofx1 ADD Xx1 ABC 333 123 x2 DEF 1903 xx3 TRY 13 x99 Aaa	x1 ADD Xx1 ABC 333 123 x2 DEF 1903 xx3 TRY 13 x99 Aaa Search Data with iterator in miliseconds :1594	As Expected	Pass
T10	Check Choice Replacement data for read iterator text valid data	A->Read Text and Construct,Iterator not used B->Add text by index, iterator not used C->Search characters then return index, iterator not used D->Replace character	Choice:H m x	Replace data with iteator in miliseconds :4618	As Expected	Pass

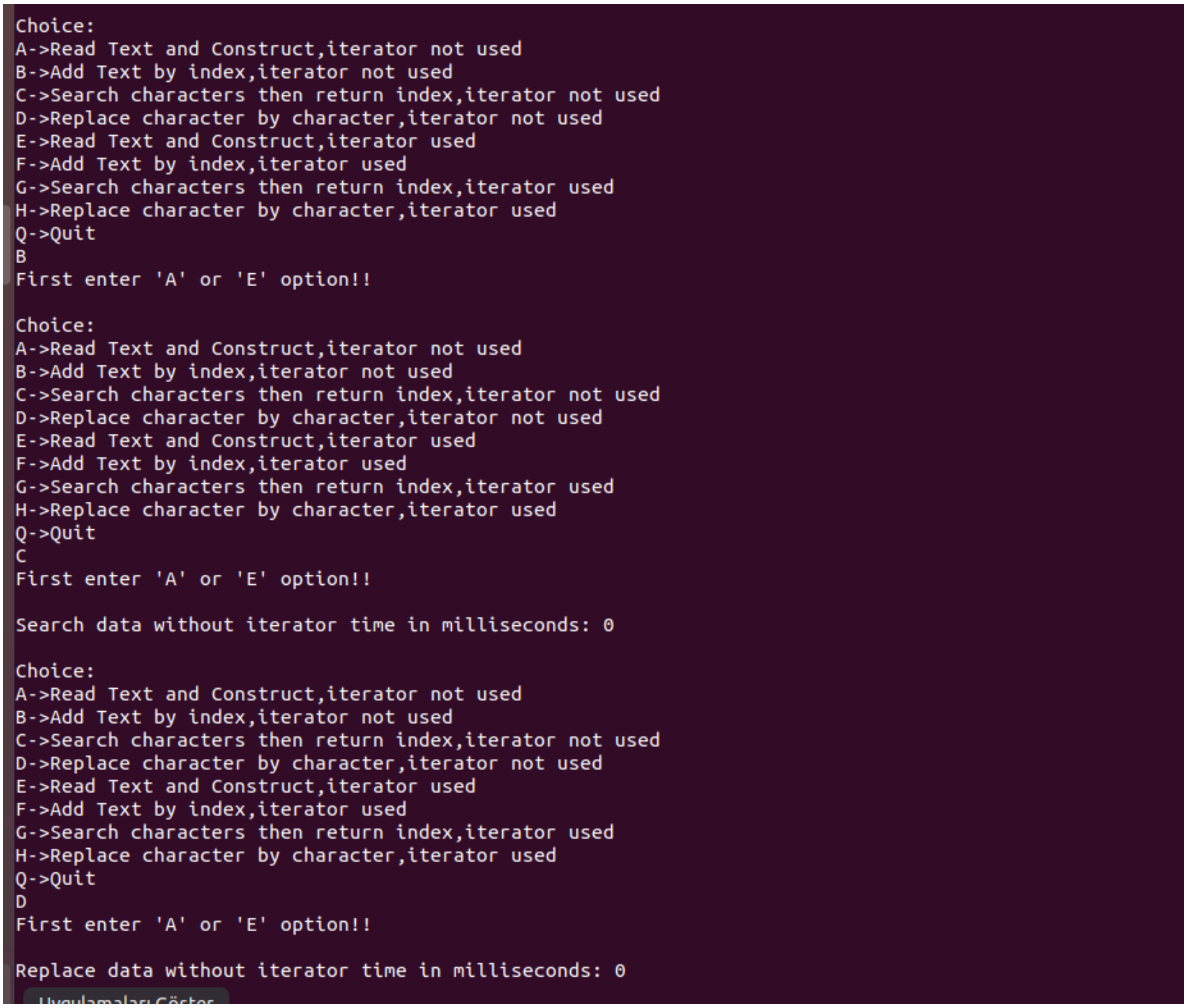
		by character, iterator not used E->Read Text and Construct,Iterator not used F->Add text by index, iterator not used G->Search characters then return index, iterator not used H->Replace character by character, iterator not used Q->Quit				
T11	Check Quit	A->Read Text and Construct,Iterator not used B->Add text by index, iterator not used C->Search characters then return index, iterator not used D->Replace character by character, iterator not used E->Read Text and Construct,Iterator not used F->Add text by index, iterator not used G->Search characters then return index, iterator not used H->Replace character by character, iterator not used Q->Quit	Choice:Q	Exit..	As Expected	Pass
T12		A->Read Text and Construct,Iterator not used B->Add text by index, iterator not used C->Search characters then return index, iterator not used D->Replace character by character, iterator not used E->Read Text and Construct,Iterator not used			As Expected	

		F->Add text by index, iterator not used G->Search characters then return index, iterator not used H->Replace character by character, iterator not used Q->Quit				
T13		A->Read Text and Construct,Iterator not used B->Add text by index, iterator not used C->Search characters then return index, iterator not used D->Replace character by character, iterator not used E->Read Text and Construct,Iterator not used F->Add text by index, iterator not used G->Search characters then return index, iterator not used H->Replace character by character, iterator not used Q->Quit			As Expected	

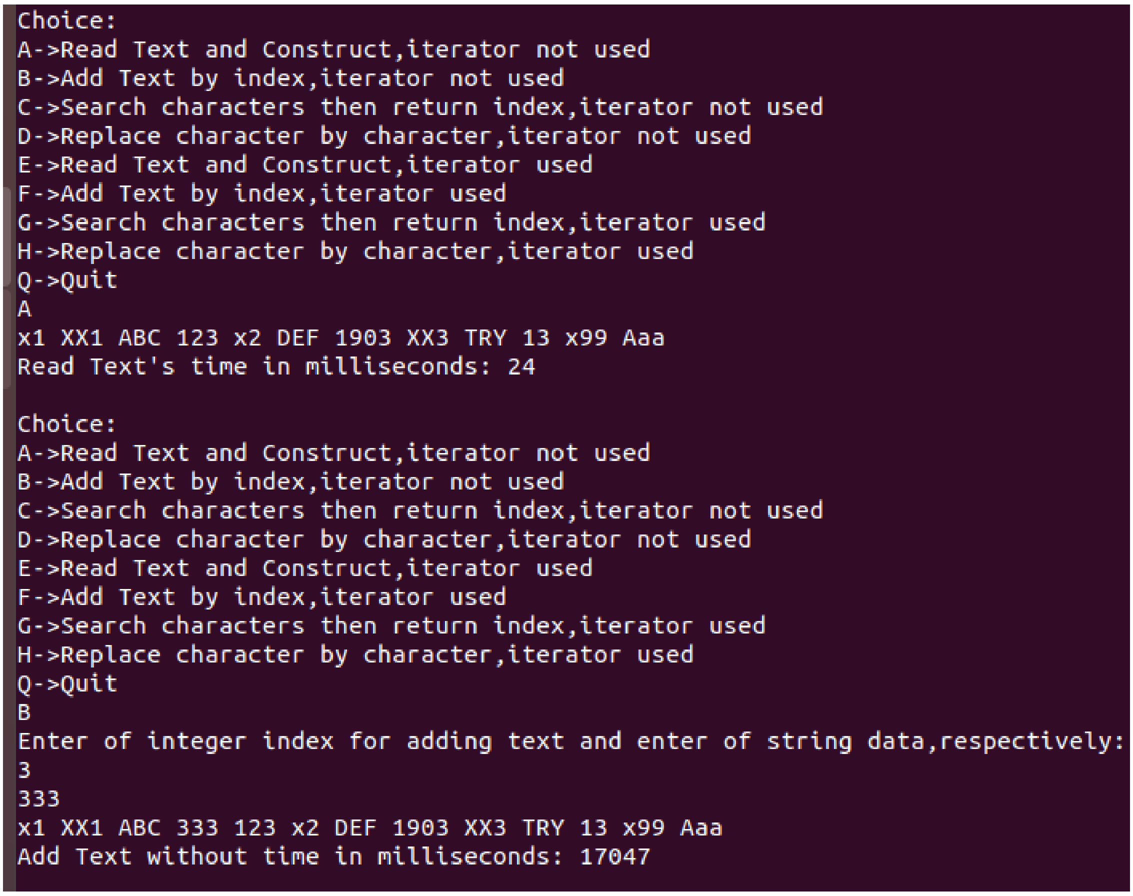
6. RUNNING AND RESULTS



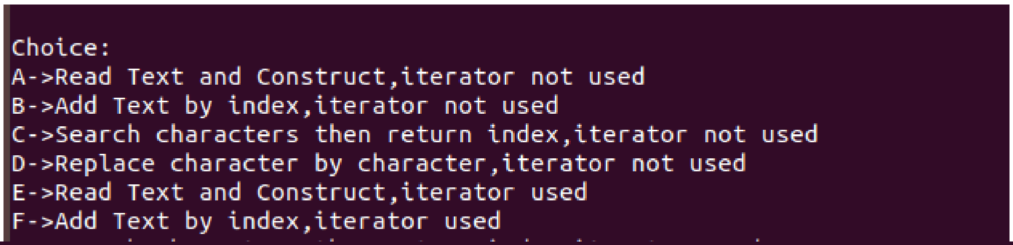
```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/List/HW2$ java Test
Nis 01, 2020 12:15:35 ÖÖ Test main
INFO: This class is Test
Two different files choice one of them,Enter 1(dosya.txt) or 2(dosya2.txt):
1

Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
A
x1 XX1 ABC 123 x2 DEF 1903 XX3 TRY 13 x99 Aaa
Read Text's time in milliseconds: 35

Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
```



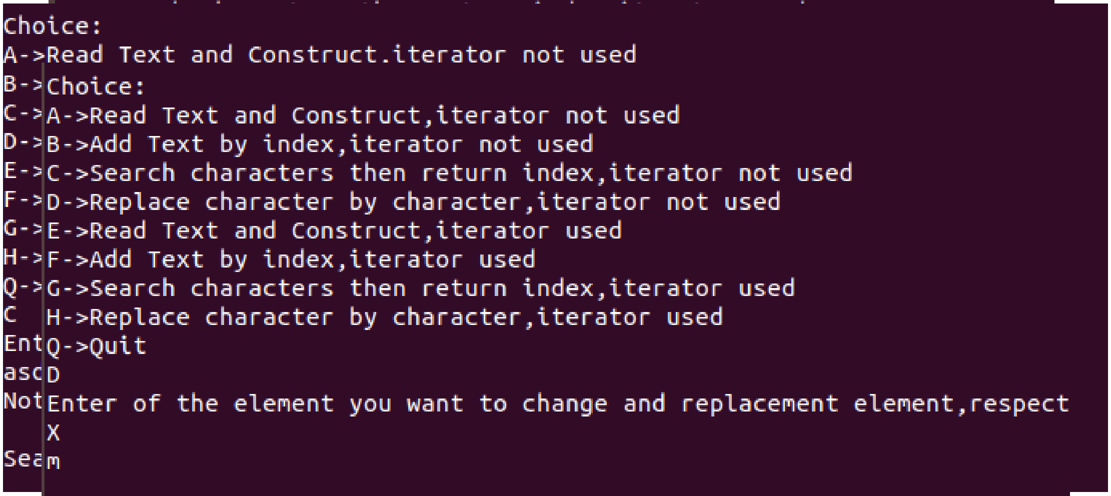
```
Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
B
First enter 'A' or 'E' option!!

Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
C
First enter 'A' or 'E' option!!

Search data without iterator time in milliseconds: 0

Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
D
First enter 'A' or 'E' option!!

Replace data without iterator time in milliseconds: 0
```



```
Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
A
x1 XX1 ABC 123 x2 DEF 1903 XX3 TRY 13 x99 Aaa
Read Text's time in milliseconds: 24

Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
B
Enter of integer index for adding text and enter of string data,respectively:
3
333
x1 XX1 ABC 333 123 x2 DEF 1903 XX3 TRY 13 x99 Aaa
Add Text without time in milliseconds: 17047
```

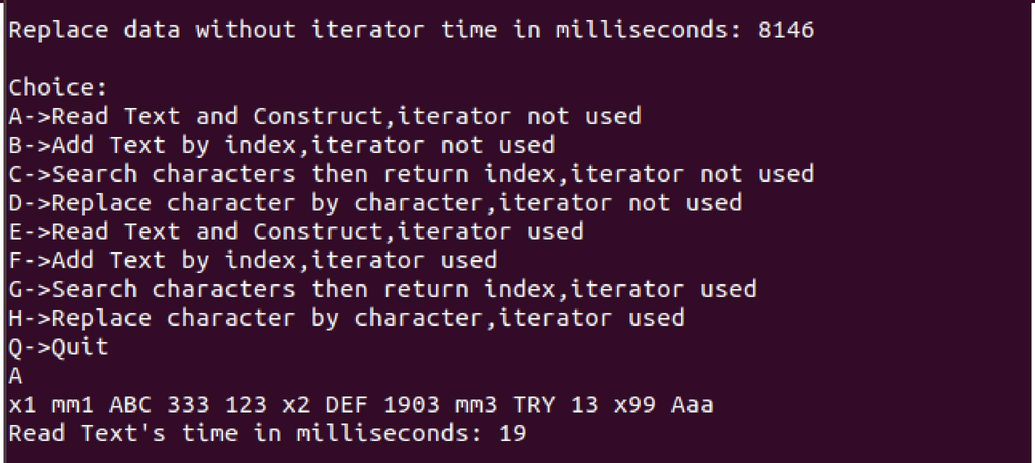


```
Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
```

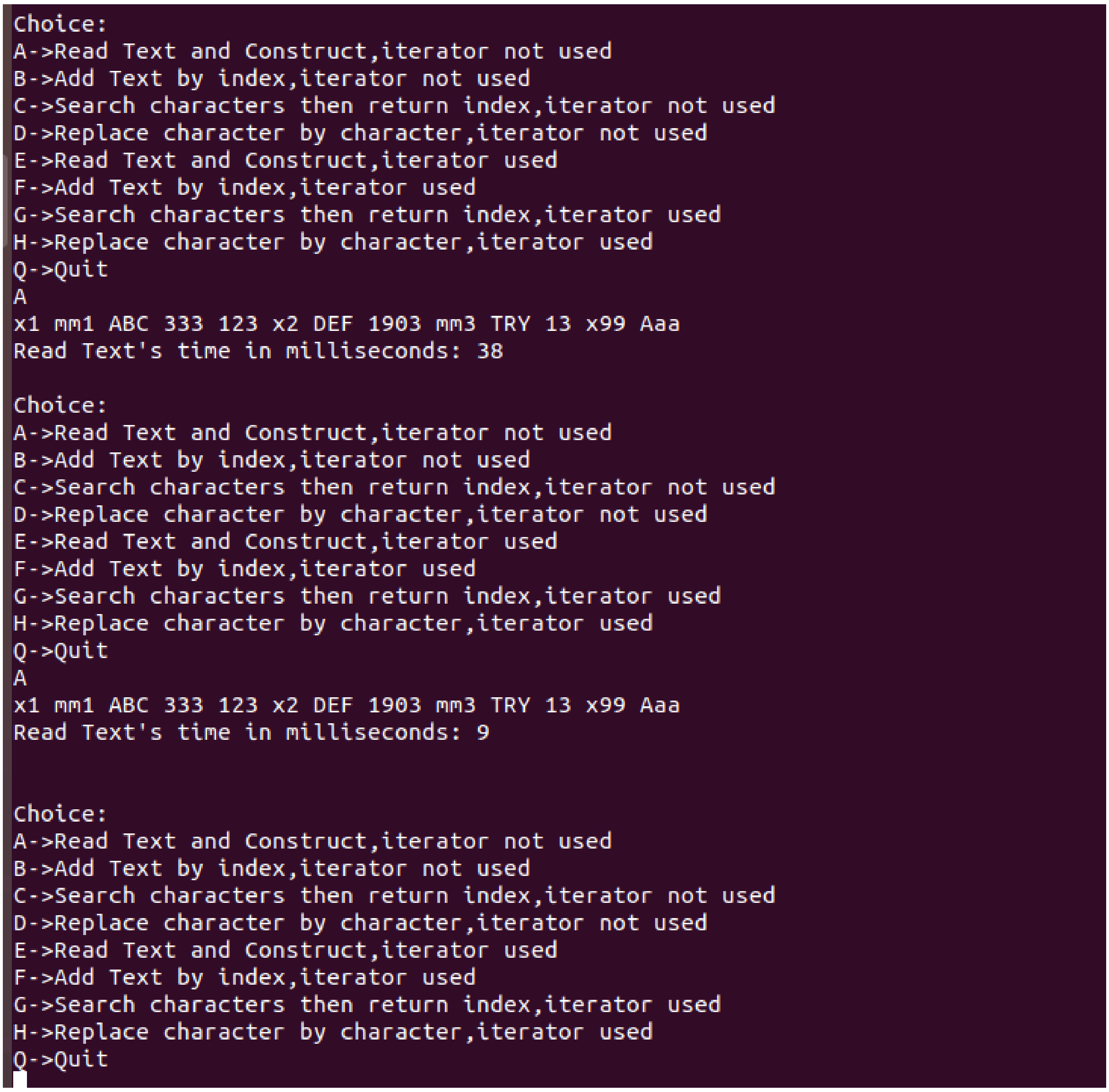


```
Choice:
A->Read Text and Construct.iterator not used
B->
C->
D->
E->
F->
G->
H->
Q->
C
Ent
asc
Not

Sea
```

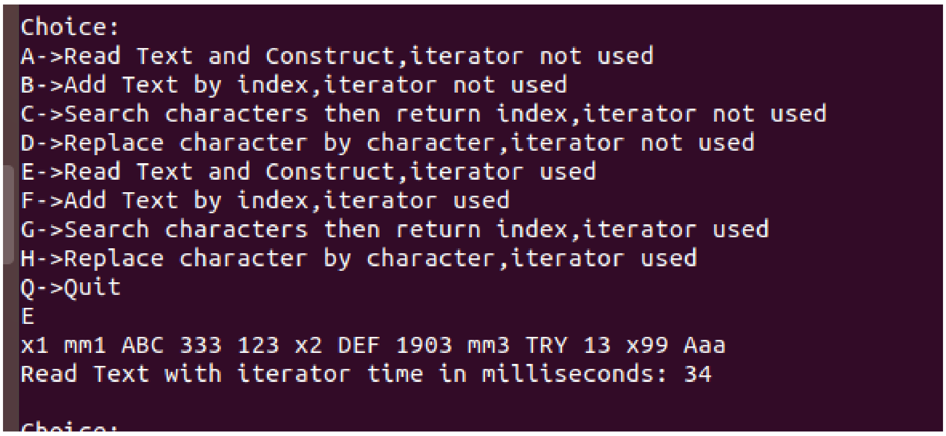
```
Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
D
Enter of the element you want to change and replacement element,respect
X
m
```



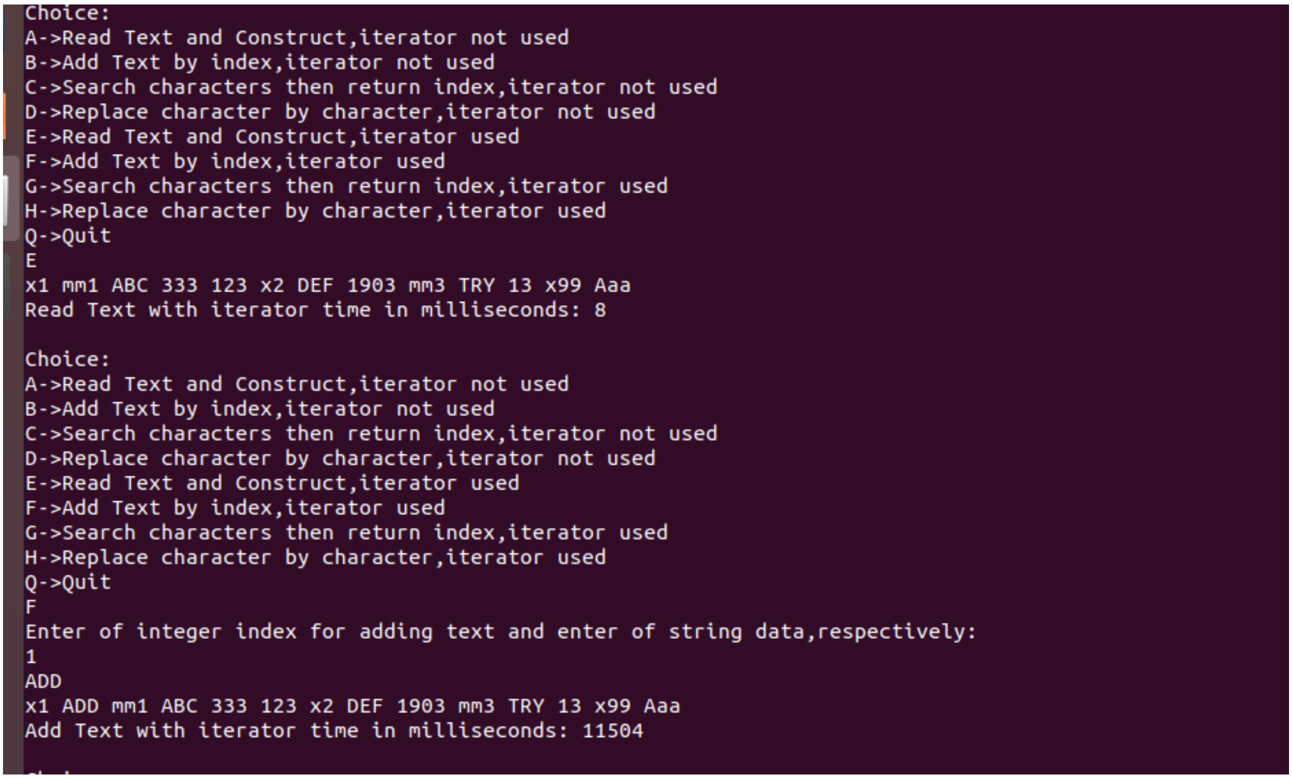
```
Replace data without iterator time in milliseconds: 8146

Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
A
x1 mm1 ABC 333 123 x2 DEF 1903 mm3 TRY 13 x99 Aaa
Read Text's time in milliseconds: 19
```



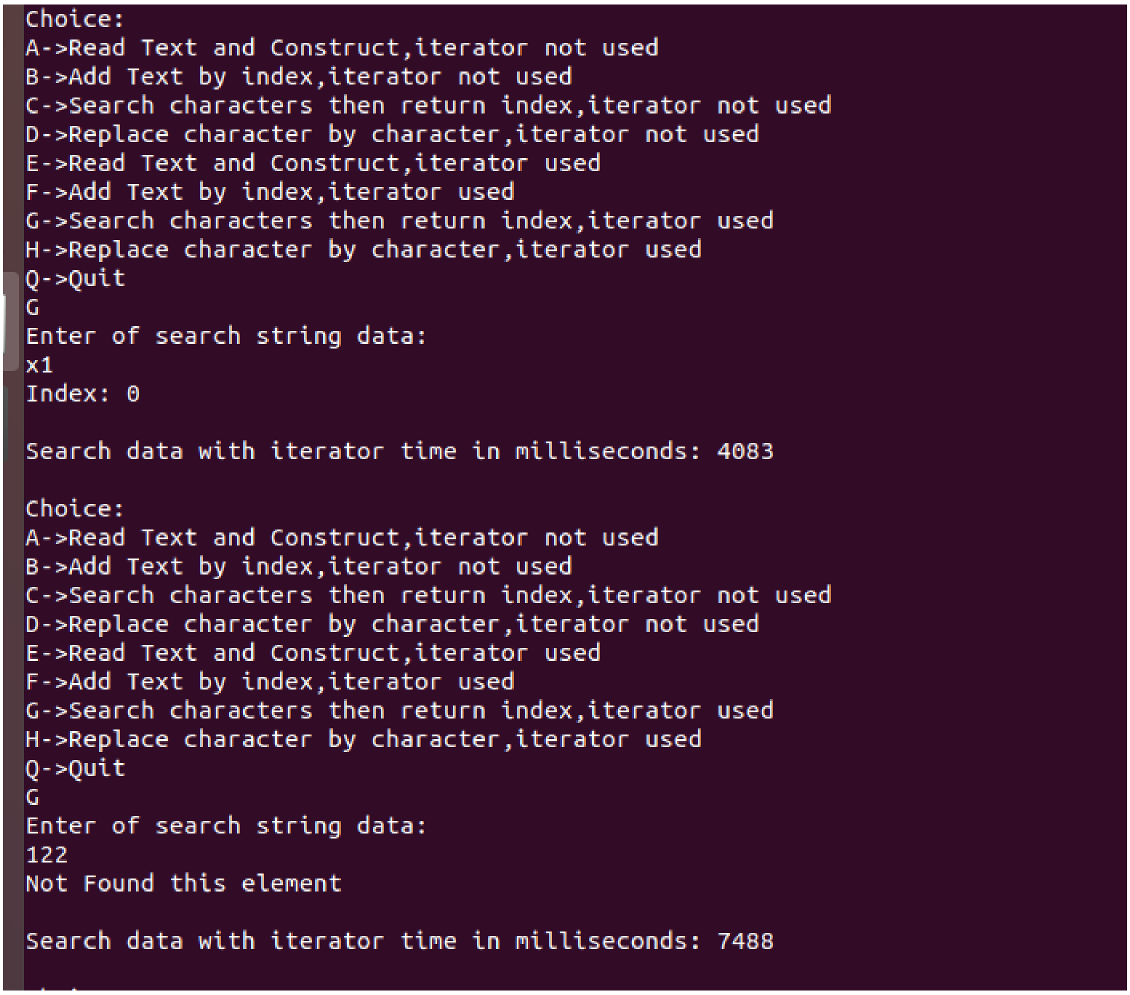
```
Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
A
x1 mm1 ABC 333 123 x2 DEF 1903 mm3 TRY 13 x99 Aaa
Read Text's time in milliseconds: 38

Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
A
x1 mm1 ABC 333 123 x2 DEF 1903 mm3 TRY 13 x99 Aaa
Read Text's time in milliseconds: 9


Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
```



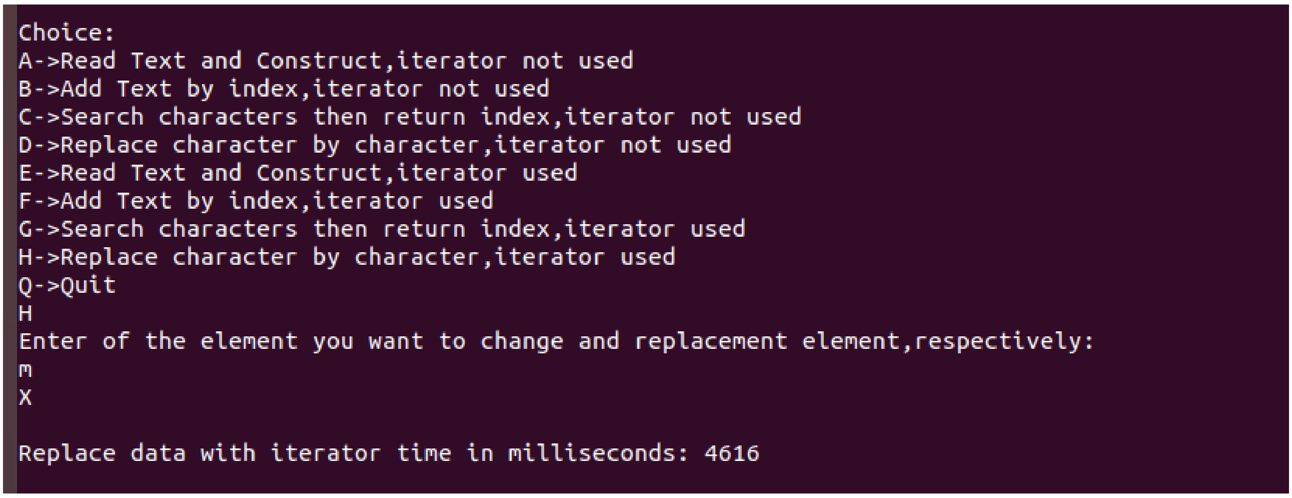
```
Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
E
x1 mm1 ABC 333 123 x2 DEF 1903 mm3 TRY 13 x99 Aaa
Read Text with iterator time in milliseconds: 34

Choice:
```



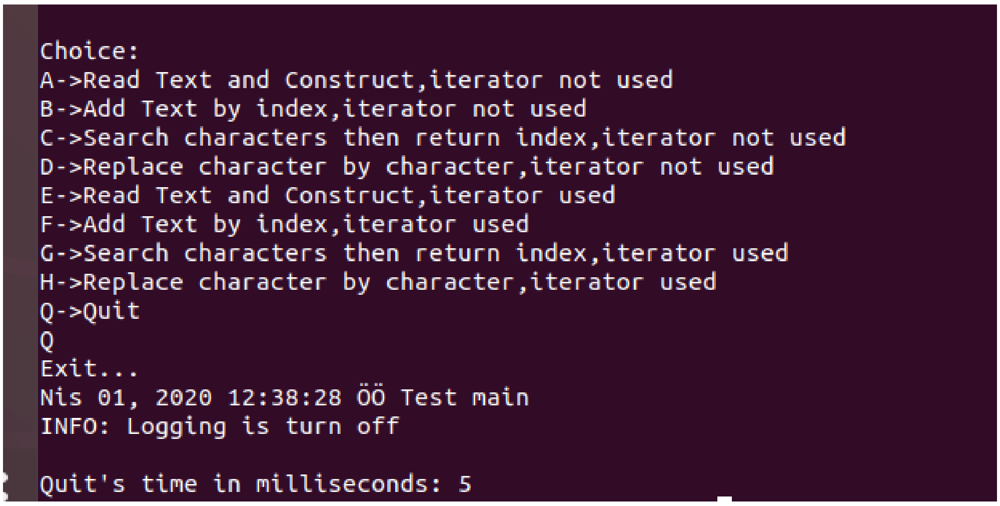
```
Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
E
x1 mm1 ABC 333 123 x2 DEF 1903 mm3 TRY 13 x99 Aaa
Read Text with iterator time in milliseconds: 8

Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
F
Enter of integer index for adding text and enter of string data,respectively:
1
ADD
x1 ADD mm1 ABC 333 123 x2 DEF 1903 mm3 TRY 13 x99 Aaa
Add Text with iterator time in milliseconds: 11504
```



```
Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
G
Enter of search string data:
x1
Index: 0

Search data with iterator time in milliseconds: 4083

Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
G
Enter of search string data:
122
Not Found this element

Search data with iterator time in milliseconds: 7488
```



```
Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
H
Enter of the element you want to change and replacement element,respectively:
m
X

Replace data with iterator time in milliseconds: 4616
```



```
Choice:
A->Read Text and Construct,iterator not used
B->Add Text by index,iterator not used
C->Search characters then return index,iterator not used
D->Replace character by character,iterator not used
E->Read Text and Construct,iterator used
F->Add Text by index,iterator used
G->Search characters then return index,iterator used
H->Replace character by character,iterator used
Q->Quit
Q
Exit...
Nis 01, 2020 12:38:28 ÖÖ Test main
INFO: Logging is turn off

Quit's time in milliseconds: 5
```