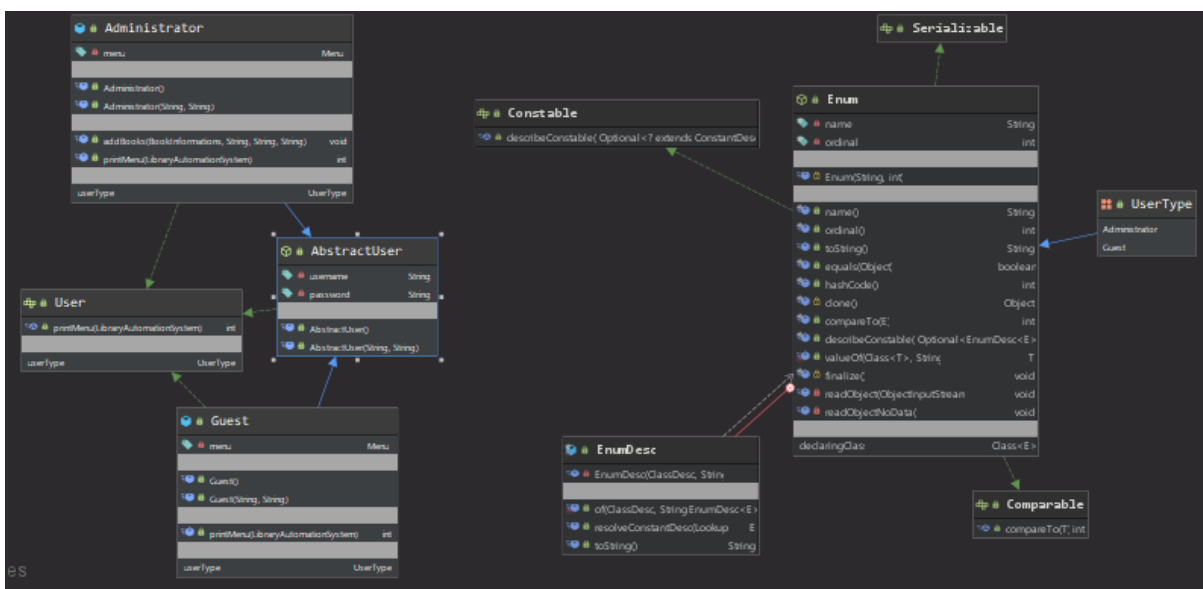
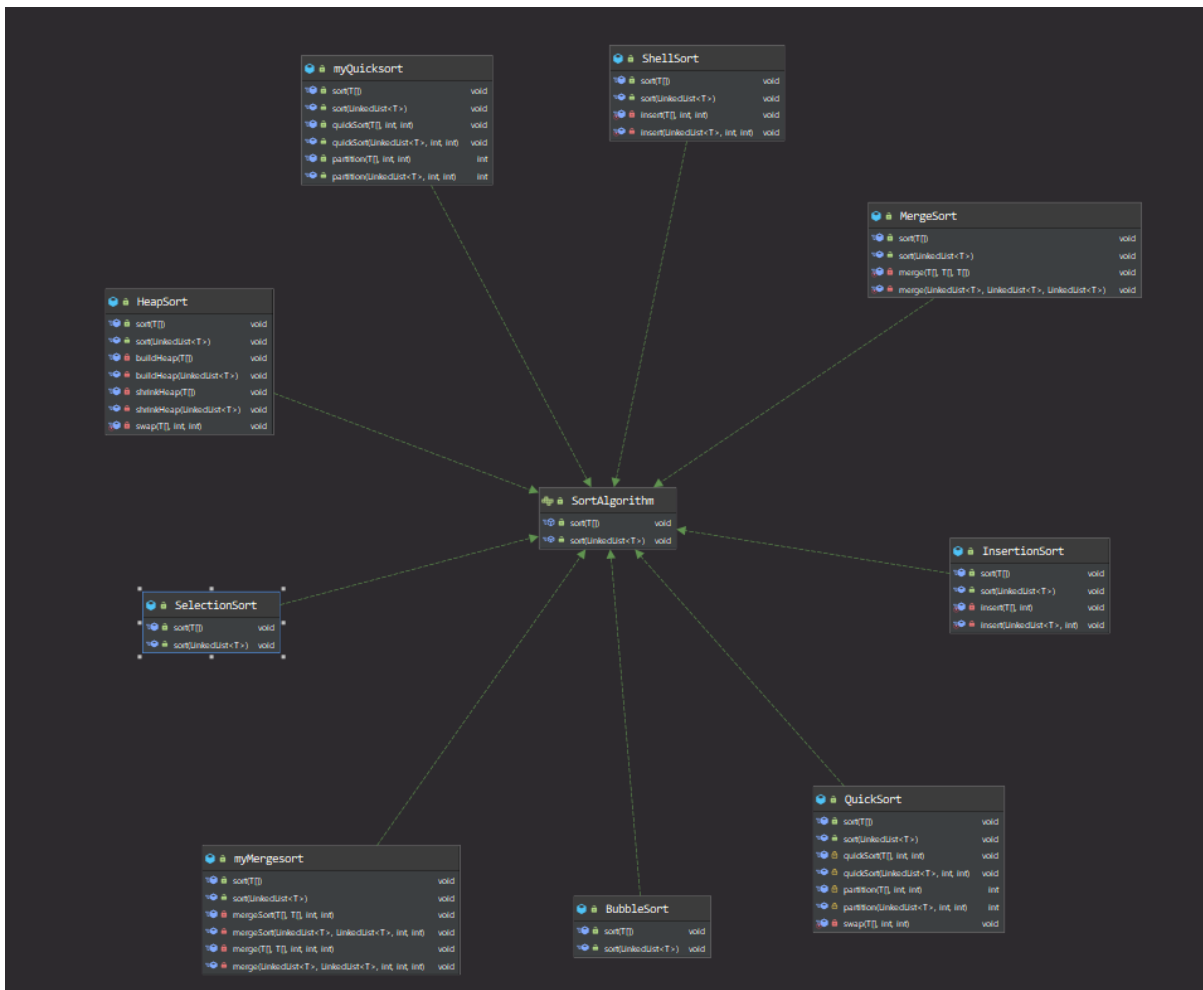
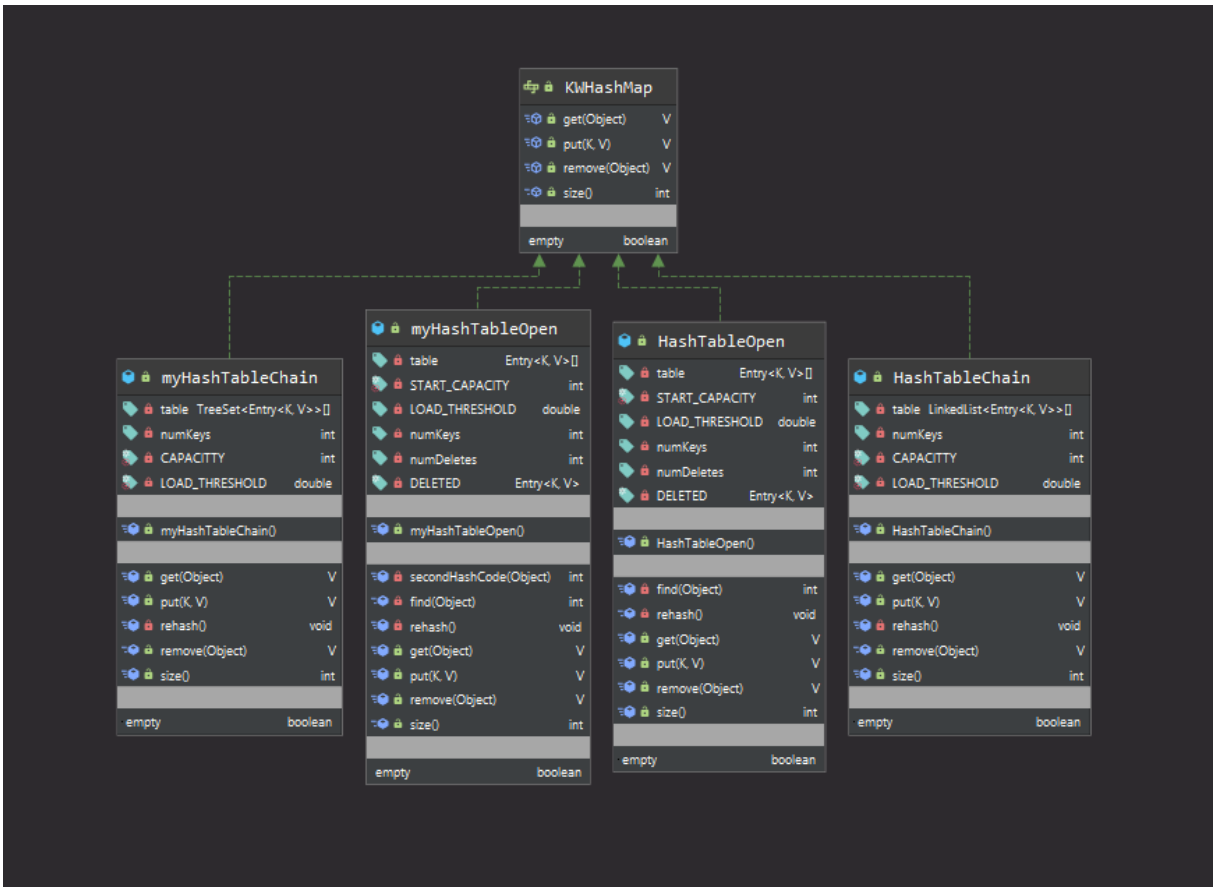
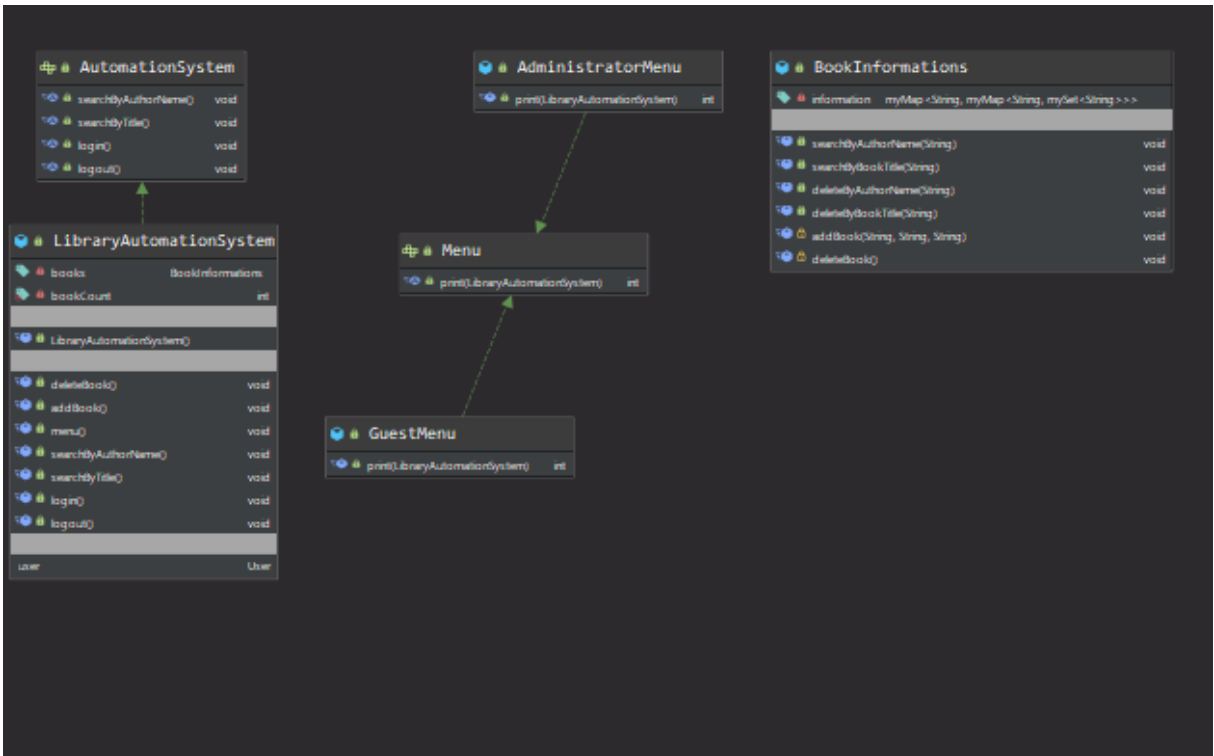


GIT Department of Computer Engineering
CSE 222/505 - Spring 2020
Homework #6 Report

Harun ALBAYRAK
171044014

1 – Class diagrams





2 – Problem solution approach

Q2) First of all, I implemented Quick sort and Merge sort. Then, i created linked lists with 1000 elements,2000 elements,3000 elements,4000 elements,5000 elements and 10000 elements. I could not create a linked list with more than 10000 elements because it takes too long to test. For testing a linked list with 10000 elements i spent approximately 1 hour. Probably I would have spent approximately 1-2 days for testing if i had created a linked list with 180000 elements because bubble sort,insertion sort,selection sort is takes too long time.

Q3) First, I created an enum that contains Administrator and Guest. I created an User interface and i created Administrator and Guest classes that implement this interface. Also these classes extend AbstractUser abstract class. I created an AutomationSystem interface. And I created LibraryAutomationSystem class that implements this interface.

Q4) First, I created myHashTableOpen and myHashTableChain classes that implement KWHashMap. Also I tested these classes in the Q4_Main.java.

3 – Test cases

Q2)

Test Scenario	Expected Results	Actual Results	Pass/Fail
Testing 9 Sort methods	Tests correctly	It has tested correctly	Pass

Q3)

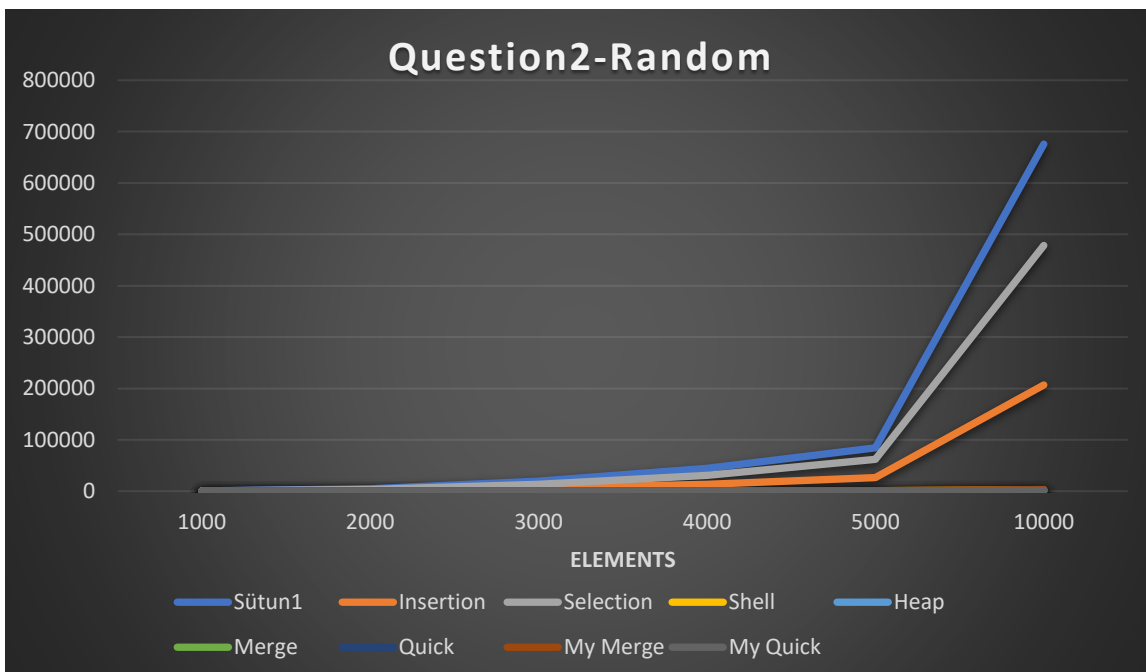
Test Scenario	Expected Results	Actual Results	Pass/Fail
Adds a Book	Adds correctly	It has added correctly	Pass
Deletes a Book	Deletes correctly	It has deleted correctly	Pass
Find a Book by an author name	Find correctly	It has found correctly	Pass
Find a Book by a title	Checks correctly	It has checked correctly	Pass
Administrator Login	Login correctly	It has login correctly	Pass
Administrator Logout	Logout correctly	It has logout correctly	Pass

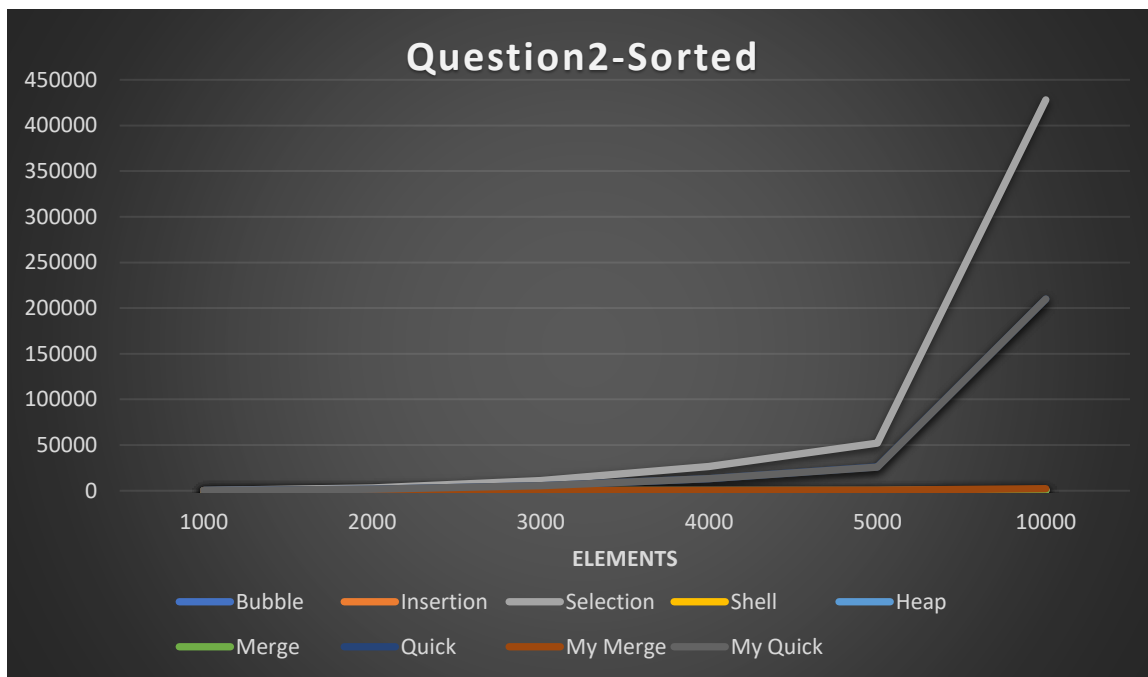
Q4)

Test Scenario	Expected Results	Actual Results	Pass/Fail
Testing classes that implement KwHashMap interface	Tests correctly	It has tested correctly	Pass

4 – Running command and results

Q2) these numbers are milisecond.





```

Bubble Sort(1000-Random): 589
Bubble Sort(1000-Sorted): 0
Insertion Sort(1000-Random): 242
Insertion Sort(1000-Sorted): 2
Selection Sort(1000-Random): 527
Selection Sort(1000-Sorted): 367
Shell Sort(1000-Random): 18
Shell Sort(1000-Sorted): 9
Heap Sort(1000-Random): 10
Heap Sort(1000-Sorted): 11
Merge Sort(1000-Random): 7
Merge Sort(1000-Sorted): 3
Quick Sort(1000-Random): 11
Quick Sort(1000-Sorted): 190
My Merge Sort(1000-Random): 24
My Merge Sort(1000-Sorted): 17
My Quick Sort(1000-Random): 13
My Quick Sort(1000-Sorted): 204

Process finished with exit code 0

```

```

Bubble Sort(2000-Random): 4870
Bubble Sort(2000-Sorted): 3
Insertion Sort(2000-Random): 1470
Insertion Sort(2000-Sorted): 5
Selection Sort(2000-Random): 3659
Selection Sort(2000-Sorted): 3200
Shell Sort(2000-Random): 80
Shell Sort(2000-Sorted): 42
Heap Sort(2000-Random): 44
Heap Sort(2000-Sorted): 44
Merge Sort(2000-Random): 16
Merge Sort(2000-Sorted): 10
Quick Sort(2000-Random): 53
Quick Sort(2000-Sorted): 1656
My Merge Sort(2000-Random): 122
My Merge Sort(2000-Sorted): 72
My Quick Sort(2000-Random): 48
My Quick Sort(2000-Sorted): 1555

```


Bubble Sort(3000-Random): 19673	Bubble Sort(4000-Random): 44270
Bubble Sort(3000-Sorted): 8	Bubble Sort(4000-Sorted): 14
Insertion Sort(3000-Random): 6602	Insertion Sort(4000-Random): 13090
Insertion Sort(3000-Sorted): 12	Insertion Sort(4000-Sorted): 21
Selection Sort(3000-Random): 13246	Selection Sort(4000-Random): 31054
Selection Sort(3000-Sorted): 11155	Selection Sort(4000-Sorted): 26553
Shell Sort(3000-Random): 212	Shell Sort(4000-Random): 380
Shell Sort(3000-Sorted): 108	Shell Sort(4000-Sorted): 184
Heap Sort(3000-Random): 100	Heap Sort(4000-Random): 168
Heap Sort(3000-Sorted): 101	Heap Sort(4000-Sorted): 171
Merge Sort(3000-Random): 28	Merge Sort(4000-Random): 46
Merge Sort(3000-Sorted): 20	Merge Sort(4000-Sorted): 36
Quick Sort(3000-Random): 157	Quick Sort(4000-Random): 247
Quick Sort(3000-Sorted): 5436	Quick Sort(4000-Sorted): 13176
My Merge Sort(3000-Random): 266	My Merge Sort(4000-Random): 468
My Merge Sort(3000-Sorted): 176	My Merge Sort(4000-Sorted): 330
My Quick Sort(3000-Random): 117	My Quick Sort(4000-Random): 231
My Quick Sort(3000-Sorted): 5498	My Quick Sort(4000-Sorted): 13177

Bubble Sort(5000-Random): 84494	Bubble Sort(10000-Random): 675688
Bubble Sort(5000-Sorted): 21	Bubble Sort(10000-Sorted): 91
Insertion Sort(5000-Random): 26736	Insertion Sort(10000-Random): 206760
Insertion Sort(5000-Sorted): 30	Insertion Sort(10000-Sorted): 124
Selection Sort(5000-Random): 61658	Selection Sort(10000-Random): 478405
Selection Sort(5000-Sorted): 52169	Selection Sort(10000-Sorted): 428011
Shell Sort(5000-Random): 631	Shell Sort(10000-Random): 2731
Shell Sort(5000-Sorted): 291	Shell Sort(10000-Sorted): 1310
Heap Sort(5000-Random): 273	Heap Sort(10000-Random): 1096
Heap Sort(5000-Sorted): 254	Heap Sort(10000-Sorted): 1029
Merge Sort(5000-Random): 63	Merge Sort(10000-Random): 223
Merge Sort(5000-Sorted): 41	Merge Sort(10000-Sorted): 176
Quick Sort(5000-Random): 409	Quick Sort(10000-Random): 1569
Quick Sort(5000-Sorted): 26054	Quick Sort(10000-Sorted): 209889
My Merge Sort(5000-Random): 728	My Merge Sort(10000-Random): 3211
My Merge Sort(5000-Sorted): 554	My Merge Sort(10000-Sorted): 2253
My Quick Sort(5000-Random): 348	My Quick Sort(10000-Random): 1490
My Quick Sort(5000-Sorted): 25973	My Quick Sort(10000-Sorted): 209918

Q3)

```
-----
-- Library Automation System Menu(Guest) --
1 - Search using author name
2 - Search using title
3 - Administrator login
(Press -1 to exit)
-----

3
Username : admin
Password : 1234
You have successfully logged in.
-----
-- Library Automation System Menu(Admin) --
1 - Search using author name
2 - Search using title
3 - Add book
4 - Delete book
5 - Administrator logout
(Press -1 to exit)
-----

3
Please enter author name: harun albayrak
Please enter title of the book: book1
Please enter shelf no: 3
Please enter corridor no: 2
The book has added.
harun albayrak - book1 - c2s3.1
```

```
-----
-- Library Automation System Menu(Admin) --
1 - Search using author name
2 - Search using title
3 - Add book
4 - Delete book
5 - Administrator logout
(Press -1 to exit)
-----
3
Please enter author name: mehmet emin karakaya
Please enter title of the book: book2
Please enter shelf no: 3
Please enter corridor no: 4
The book has added.
mehmet emin karakaya - book2 - c4s3.2
-----
-- Library Automation System Menu(Admin) --
1 - Search using author name
2 - Search using title
3 - Add book
4 - Delete book
5 - Administrator logout
(Press -1 to exit)
-----
3
Please enter author name: mehmet emin karakaya
Please enter title of the book: book3
Please enter shelf no: 4
Please enter corridor no: 5
The book has added.
mehmet emin karakaya - book3 - c5s4.3
-----
```

-- Library Automation System Menu(Admin) --

1 - Search using author name

2 - Search using title

3 - Add book

4 - Delete book

5 - Administrator logout

(Press -1 to exit)

3

Please enter author name: *mehmet emin karakaya*

Please enter title of the book: *book3*

Please enter shelf no: *4*

Please enter corridor no: *6*

The book has added.

mehmet emin karakaya - book3 - c6s4.4

-- Library Automation System Menu(Admin) --

1 - Search using author name

2 - Search using title

3 - Add book

4 - Delete book

5 - Administrator logout

(Press -1 to exit)

1

Please enter an author name you want to search:

mehmet emin karakaya

0 - book2

1 - book3

Please choose number to display its location.(-1 to exit)

1

Author Name: mehmet emin karakaya ---- Location: c5s4.3 - c6s4.4

```
-----
-- Library Automation System Menu(Admin) --
1 - Search using author name
2 - Search using title
3 - Add book
4 - Delete book
5 - Administrator logout
(Press -1 to exit)
-----

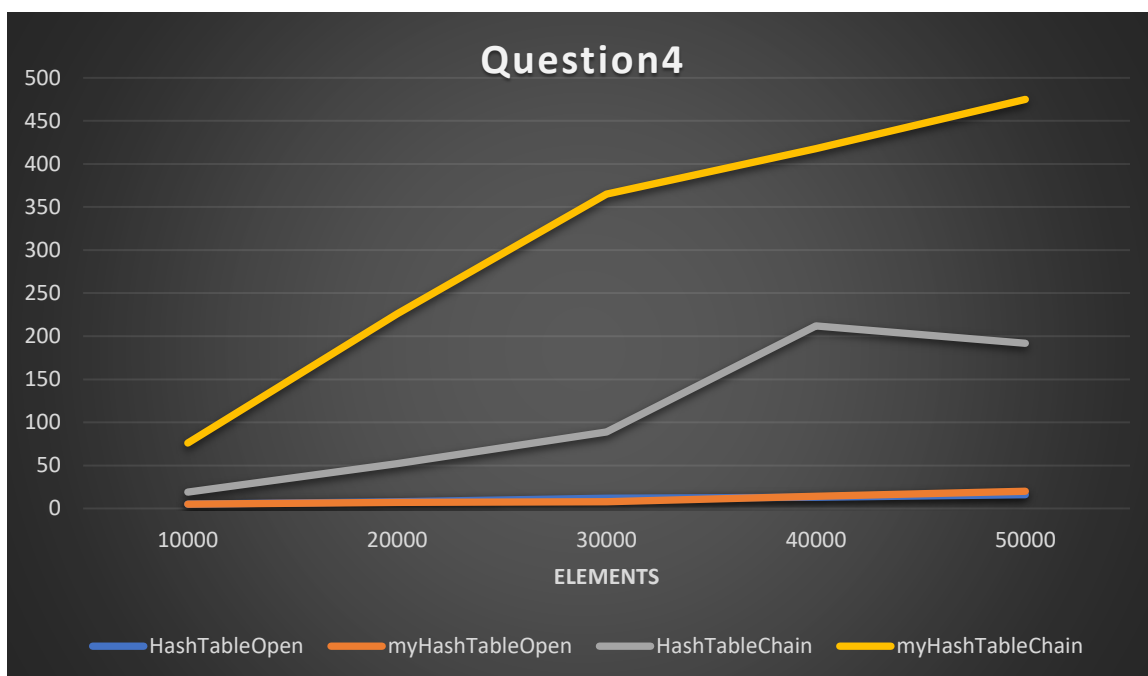
2
Please enter title of the book you want to search:
book1
Author Name: harun albayrak ---- Location: c2s3.1
-----
-- Library Automation System Menu(Admin) --
1 - Search using author name
2 - Search using title
3 - Add book
4 - Delete book
5 - Administrator logout
(Press -1 to exit)
-----

4
0. harun albayrak
1. mehmet emin karakaya
Which author's book would you like to delete?
1
0 - book2
1 - book3
Please choose number to delete the book.(-1 to exit)
1
The book has deleted.
```

Q4) these numbers are milisecond.

put	10000 Elements	20000 Elements	30000 Element	40000 Elements	50000 Elements
Hash Table Open(Linear Probing)	5	8	12	13	16
My Hash Table Open(Double Hashing)	5	7	8	14	20
Hash Table Chain(LinkedList)	19	52	89	212	192
My Hash Table Chain(TreeSet)	76	226	365	418	475

remove	10000 Elements	20000 Elements	30000 Element	40000 Elements	50000 Elements
Hash Table Open(Linear Probing)	0	1	2	2	3
My Hash Table Open(Double Hashing)	1	1	2	2	2
Hash Table Chain(LinkedList)	1	1	2	2	2
My Hash Table Chain(TreeSet)	1	1	1	1	2



```
Hash Map Open(Linear Probing-Put Elements-10000): 5
My Hash Map Open(Double Hashing-Put Elements-10000): 5
Hash Map Chain(LinkedList-Put Elements-10000): 19
My Hash Map Chain(TreeSet-Put Elements-10000): 76
-----
Hash Map Open(Linear Probing-Remove Elements-10000): 0
My Hash Map Open(Double Hashing-Remove Elements-10000): 1
Hash Map Chain(LinkedList-Remove Elements-10000): 1
My Hash Map Chain(TreeSet-Remove Elements-10000): 1
```

```
Hash Map Open(Linear Probing-Put Elements-20000): 8
My Hash Map Open(Double Hashing-Put Elements-20000): 7
Hash Map Chain(LinkedList-Put Elements-20000): 52
My Hash Map Chain(TreeSet-Put Elements-20000): 226
-----
Hash Map Open(Linear Probing-Remove Elements-20000): 1
My Hash Map Open(Double Hashing-Remove Elements-20000): 1
Hash Map Chain(LinkedList-Remove Elements-20000): 1
My Hash Map Chain(TreeSet-Remove Elements-20000): 1
```

```
Hash Map Open(Linear Probing-Put Elements-30000): 12
My Hash Map Open(Double Hashing-Put Elements-30000): 8
Hash Map Chain(LinkedList-Put Elements-30000): 89
My Hash Map Chain(TreeSet-Put Elements-30000): 365
-----
Hash Map Open(Linear Probing-Remove Elements-30000): 2
My Hash Map Open(Double Hashing-Remove Elements-30000): 2
Hash Map Chain(LinkedList-Remove Elements-30000): 2
My Hash Map Chain(TreeSet-Remove Elements-30000): 1
```

```
Hash Map Open(Linear Probing-Put Elements-40000): 13
My Hash Map Open(Double Hashing-Put Elements-40000): 14
Hash Map Chain(LinkedList-Put Elements-40000): 212
My Hash Map Chain(TreeSet-Put Elements-40000): 418
-----
Hash Map Open(Linear Probing-Remove Elements-40000): 2
My Hash Map Open(Double Hashing-Remove Elements-40000): 2
Hash Map Chain(LinkedList-Remove Elements-40000): 2
My Hash Map Chain(TreeSet-Remove Elements-40000): 1
```

```
Hash Map Open(Linear Probing-Put Elements-50000): 16
My Hash Map Open(Double Hashing-Put Elements-50000): 20
Hash Map Chain(LinkedList-Put Elements-50000): 192
My Hash Map Chain(TreeSet-Put Elements-50000): 475
-----
Hash Map Open(Linear Probing-Remove Elements-50000): 3
My Hash Map Open(Double Hashing-Remove Elements-50000): 2
Hash Map Chain(LinkedList-Remove Elements-50000): 2
My Hash Map Chain(TreeSet-Remove Elements-50000): 2
```