

# Output (Lab pgm 1)

Sunday, November 5, 2017 9:59 PM

```
C:\WINDOWS\system32\cmd.exe

E:\nitk\semester1\dataStructures\Labpgms\Labpgm1>javac -Xlint:unchecked -source
ath linearList -cp linearList linearList/*.java

E:\nitk\semester1\dataStructures\Labpgms\Labpgm1>jar -cvmf manifest.txt linear
st.jar linearList
added manifest
adding: linearList/(in = 0) (out= 0)(stored 0%)
adding: linearList/ArrayLinearList.class(in = 2492) (out= 1270)(deflated 49%)
adding: linearList/ArrayLinearList.java(in = 3372) (out= 1128)(deflated 66%)
adding: linearList/ArrayMain.class(in = 1536) (out= 784)(deflated 48%)
adding: linearList/ArrayMain.java(in = 750) (out= 307)(deflated 59%)
adding: linearList/LinearList.class(in = 583) (out= 326)(deflated 44%)
adding: linearList/LinearList.java(in = 328) (out= 174)(deflated 46%)
adding: linearList/LinkedList.class(in = 2173) (out= 1195)(deflated 45%)
adding: linearList/LinkedList.java(in = 2863) (out= 920)(deflated 67%)
adding: linearList/LinkedListMain.class(in = 1506) (out= 762)(deflated 49%)
adding: linearList/LinkedListMain.java(in = 731) (out= 301)(deflated 58%)
adding: linearList/Node.class(in = 708) (out= 358)(deflated 49%)
adding: linearList/Node.java(in = 288) (out= 153)(deflated 46%)

E:\nitk\semester1\dataStructures\Labpgms\Labpgm1>java -jar linearList.jar
1 2 3 4
list after adding 5 at index 2 : 1 2 5 3 4
list after removing element 3 from the list : 1 2 5 4
IndexOf 5 = 2
list after adding 100 at index 2 : 1 2 100 5 4
getI(4) = 4
list after adding element at position 5 : 1 2 100 5 4 200

E:\nitk\semester1\dataStructures\Labpgms\Labpgm1>
```

```
C:\WINDOWS\system32\cmd.exe

E:\nitk\semester1\dataStructures\Labpgms\Labpgm1>javac -Xlint:unchecked -sourc
ath linearList -cp linearList linearList/*.java

E:\nitk\semester1\dataStructures\Labpgms\Labpgm1>jar -cvmf manifest2.txt linke
ist.jar linearList
added manifest
adding: linearList/(in = 0) (out= 0)(stored 0%)
adding: linearList/ArrayLinearList.class(in = 2492) (out= 1270)(deflated 49%)
adding: linearList/ArrayLinearList.java(in = 3372) (out= 1128)(deflated 66%)
adding: linearList/ArrayMain.class(in = 1536) (out= 784)(deflated 48%)
adding: linearList/ArrayMain.java(in = 750) (out= 307)(deflated 59%)
adding: linearList/LinearList.class(in = 583) (out= 326)(deflated 44%)
adding: linearList/LinearList.java(in = 328) (out= 174)(deflated 46%)
adding: linearList/LinkedList.class(in = 2173) (out= 1195)(deflated 45%)
adding: linearList/LinkedList.java(in = 2863) (out= 920)(deflated 67%)
adding: linearList/LinkedListMain.class(in = 1506) (out= 762)(deflated 49%)
adding: linearList/LinkedListMain.java(in = 731) (out= 301)(deflated 58%)
adding: linearList/Node.class(in = 708) (out= 358)(deflated 49%)
adding: linearList/Node.java(in = 288) (out= 153)(deflated 46%)

E:\nitk\semester1\dataStructures\Labpgms\Labpgm1>java -jar linkedList.jar
1 2 3 4
list after adding 5 at index 2 1 2 3 4 5
list after removing element 3 from the list 1 2 4 5
IndexOf 5 = 3
list after adding 100 at index 2 1 2 4 5 100
getI(4) = 100
list after adding element at position 5 : 1 2 4 5 100 200

E:\nitk\semester1\dataStructures\Labpgms\Labpgm1>
```

# Output (Lab pgm 1b)

Sunday, November 5, 2017 10:03 PM

```
C:\WINDOWS\system32\cmd.exe

E:\nltk\semester1\dataStructures\Labpgms\stack\src>java -jar linearStack.jar

Enter the choice
1-push
2-pop
3-peek
4-display
5-Exit

1
Entered choice = 1

Enter the item to be pushed
50
top=0
50

Enter the choice
1-push
2-pop
3-peek
4-display
5-Exit

1
Entered choice = 1

Enter the item to be pushed
60
top=1
60 50

Enter the choice
1-push
2-pop
3-peek
4-display
5-Exit

2
Entered choice = 2

Popped element = 60
top=0
50

Enter the choice
1-push
2-pop
3-peek
4-display
5-Exit

5
Entered choice = 5

E:\nltk\semester1\dataStructures\Labpgms\stack\src>
```

```
C:\WINDOWS\system32\cmd.exe

E:\nitk\semester1\dataStructures\Labpgms\stack\src>java -jar linkedList.jar
Enter the choice
1.push
2.pop
3.peek
4.Display
5.Exit
1
Enter the element to be pushed
50
Linked Stack: [50]
Enter the choice
1.push
2.pop
3.peek
4.Display
5.Exit
1
Enter the element to be pushed
60
Linked Stack: [60->50]
Enter the choice
1.push
2.pop
3.peek
4.Display
5.Exit
2
Popped element = 60
size = 1
Linked Stack: [50]
Enter the choice
1.push
2.pop
3.peek
4.Display
5.Exit
3
Peek element = 50
Linked Stack: [50]
Enter the choice
1.push
2.pop
3.peek
4.Display
5.Exit
4
-----
Linked Stack: [50]
-----
Enter the choice
1.push
2.pop
3.peek
4.Display
5.Exit
5
E:\nitk\semester1\dataStructures\Labpgms\stack\src>
```

# Output (Lab pgm 1c)

Sunday, November 5, 2017 10:06 PM

```
C:\WINDOWS\system32\cmd.exe

E:\nitk\semester1\dataStructures\Labpgms\queue\linearQueue\src>java -jar linearQueue.jar
front=0 rear=-1 queue contents = []
front=0 rear=-1 queue contents = []

1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter the choice
1

Enter the element to be inserted
1
After enqueueing front=0 rear=0 queue contents = [1]

1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter the choice
1

Enter the element to be inserted
5
After enqueueing front=0 rear=1 queue contents = [1, 5]

1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter the choice
2

Deleted element = 1
front=1 rear=1 queue contents = [5]

1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter the choice
3

front=1 rear=1 queue contents = [5]

1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter the choice
4

E:\nitk\semester1\dataStructures\Labpgms\queue\linearQueue\src>
```

```
C:\WINDOWS\system32\cmd.exe
E:\nitk\semester1\dataStructures\Labpgms\queue\linkedQueue\src>javac -sourcepath
queue -cp queue;queue/linkedQueue;. queue/linkedQueue/*.java
E:\nitk\semester1\dataStructures\Labpgms\queue\linkedQueue\src>jar -cvmf manifes
t.txt linkedQueue.jar queue
added manifest
adding: queue/<in = 0> <out= 0><stored 0%>
adding: queue/linkedQueue/<in = 0> <out= 0><stored 0%>
adding: queue/linkedQueue/LinkedList.class<in = 3146> <out= 1682><deflated 46%>
adding: queue/linkedQueue/LinkedList.java<in = 2733> <out= 889><deflated 67%>
adding: queue/linkedQueue/Node.class<in = 575> <out= 312><deflated 45%>
adding: queue/linkedQueue/Node.java<in = 305> <out= 149><deflated 51%>
adding: queue/QueueEmptyException.class<in = 221> <out= 170><deflated 23%>
adding: queue/QueueEmptyException.java<in = 75> <out= 66><deflated 12%>
E:\nitk\semester1\dataStructures\Labpgms\queue\linkedQueue\src>java -jar linkedQ
ueue.jar
Enter the choice
enqueue-1
dequeue-2
display-3
Exit-4
1
Enter the element to be inserted into the queue
10
Enter the choice
enqueue-1
dequeue-2
display-3
Exit-4
1
Enter the element to be inserted into the queue
50
Enter the choice
enqueue-1
dequeue-2
display-3
Exit-4
3
-----10->50
Enter the choice
enqueue-1
dequeue-2
display-3
Exit-4
2
Deleted element = 10
Enter the choice
enqueue-1
dequeue-2
display-3
Exit-4
3
-----50
Enter the choice
enqueue-1
dequeue-2
display-3
Exit-4
4
```

# Output (Lab pgm 1d)

Sunday, November 5, 2017 10:07 PM

```
C:\WINDOWS\system32\cmd.exe

3.Delete at the beginning of the list
4.Delete at the end of the list
5.Display
6.Exit
Enter the choice
6

E:\nitk\semester1\dataStructures\Labpgms\circularList\src>java -jar circularList.jar
1.Insert at the beginning of the list
2.Insert at the end of the list
3.Delete at the beginning of the list
4.Delete at the end of the list
5.Display
6.Exit
Enter the choice
1
Enter the element to be inserted
Enter the element
10
[10]
1.Insert at the beginning of the list
2.Insert at the end of the list
3.Delete at the beginning of the list
4.Delete at the end of the list
5.Display
6.Exit
Enter the choice
2
Enter the element to be inserted
Enter the element
50
[10->50]
1.Insert at the beginning of the list
2.Insert at the end of the list
3.Delete at the beginning of the list
4.Delete at the end of the list
5.Display
6.Exit
Enter the choice
3
[50]
1.Insert at the beginning of the list
2.Insert at the end of the list
3.Delete at the beginning of the list
4.Delete at the end of the list
5.Display
6.Exit
Enter the choice
4
[]
1.Insert at the beginning of the list
2.Insert at the end of the list
3.Delete at the beginning of the list
4.Delete at the end of the list
5.Display
6.Exit
Enter the choice
6

E:\nitk\semester1\dataStructures\Labpgms\circularList\src>
```

# Output (Lab pgm 1e)

Sunday, November 5, 2017 10:08 PM

```
C:\WINDOWS\system32\cmd.exe
E:\nitk\semester1\dataStructures\Labpgms\doublyLinkedList\src>java -jar doublyLi
nkedList.jar
1.Insert at the beginning of the list
2.Insert at the end of the list
3.Insert at the middle of the list
4.Delete First node
5.Delete Last node
6.Delete node at middle
7.Display
Enter the choice
1
Enter the element to be inserted
10
[10]
1.Insert at the beginning of the list
2.Insert at the end of the list
3.Insert at the middle of the list
4.Delete First node
5.Delete Last node
6.Delete node at middle
7.Display
Enter the choice
3
Enter the element to be inserted
40
Enter the position to be inserted
1
[10<=>40]
1.Insert at the beginning of the list
2.Insert at the end of the list
3.Insert at the middle of the list
4.Delete First node
5.Delete Last node
6.Delete node at middle
7.Display
Enter the choice
6
Enter the position of the node to be deleted
1
[10]
1.Insert at the beginning of the list
2.Insert at the end of the list
3.Insert at the middle of the list
4.Delete First node
5.Delete Last node
6.Delete node at middle
7.Display
Enter the choice
7
[10]
1.Insert at the beginning of the list
2.Insert at the end of the list
3.Insert at the middle of the list
4.Delete First node
5.Delete Last node
6.Delete node at middle
7.Display
Enter the choice
8
E:\nitk\semester1\dataStructures\Labpgms\doublyLinkedList\src>
```

## Output (Lab pgm 2)

Sunday, November 5, 2017 10:13 PM

```
C:\WINDOWS\system32\cmd.exe

E:\nitk\semester1\dataStructures\Labpgms\polynomial\src>java -jar polynomialAddAndSubtract.jar
=====
Polynomial addition
=====
[1x^2 + 2x^1 + 1x^0]
[1x^1 + 1x^0]
[1x^2 + 3x^1 + 2x^0]

=====
Polynomial subtraction
=====
[1x^2 + 2x^1 + 1x^0]
[1x^1 + 1x^0]
[1x^2 + 1x^1 + 0x^0]

=====
Polynomial Multiplication
=====
[1x^2 + 2x^1 + 1x^0]
[1x^1 + 1x^0]
[1x^3 + 3x^2 + 3x^1 + 1x^0]

=====
Polynomial Division
=====
[1x^2 + 2x^1 + 1x^0]
[1x^1 + 1x^0]
partial remainder = [1x^2 + 2x^1 + 1x^0]
list1 = [1x^2 + 2x^1 + 1x^0]
3
List is empty
Remainder = []
Quotient = [1x^1 + 1x^0]

E:\nitk\semester1\dataStructures\Labpgms\polynomial\src>
```



# Output (Lab pgm 3)

Sunday, November 5, 2017 10:15 PM

```
C:\WINDOWS\system32\cmd.exe
E:\nith\semester1\dataStructures\Labpgms\Labpgm3\multipleStacks\src>java -jar multipleStacks.jar

Enter the choice
1.Push(Stack1)
2.Pop(Stack1)
3.Push(Stack2)
4.Pop(Stack2)
5.Display Multiple stacks
6.Exit
1
Enter the element to be pushed
10
[10 ***** ]

Enter the choice
1.Push(Stack1)
2.Pop(Stack1)
3.Push(Stack2)
4.Pop(Stack2)
5.Display Multiple stacks
6.Exit
1
Enter the element to be pushed
50
[10, 50 ***** ]

Enter the choice
1.Push(Stack1)
2.Pop(Stack1)
3.Push(Stack2)
4.Pop(Stack2)
5.Display Multiple stacks
6.Exit
3
Enter the element to be pushed
100
[10, 50 ***** 100]

Enter the choice
1.Push(Stack1)
2.Pop(Stack1)
3.Push(Stack2)
4.Pop(Stack2)
5.Display Multiple stacks
6.Exit
3
Enter the element to be pushed
150
[10, 50 ***** 150, 100]

Enter the choice
1.Push(Stack1)
2.Pop(Stack1)
3.Push(Stack2)
4.Pop(Stack2)
5.Display Multiple stacks
6.Exit
5
=====
```

# Output (Lab pgm 3b)

Sunday, November 5, 2017 10:15 PM

```
C:\WINDOWS\system32\cmd.exe
E:\nitk\semester1\dataStructures\Labpgms\Labpgm3\multipleQueues\src>java -jar multipleQueues.jar
1.Enqueue<Queue1>
2.Dequeue<Queue1>
3.Enqueue<Queue2>
4.Dequeue<Queue2>
5.Display
6.Exit
Enter the choice
1
Enter the element to be enqueued
1
[1*****]
1.Enqueue<Queue1>
2.Dequeue<Queue1>
3.Enqueue<Queue2>
4.Dequeue<Queue2>
5.Display
6.Exit
Enter the choice
1
Enter the element to be enqueued
5
[1 5*****]
1.Enqueue<Queue1>
2.Dequeue<Queue1>
3.Enqueue<Queue2>
4.Dequeue<Queue2>
5.Display
6.Exit
Enter the choice
3
Enter the element to be enqueued
100
[1 5*****100]
1.Enqueue<Queue1>
2.Dequeue<Queue1>
3.Enqueue<Queue2>
4.Dequeue<Queue2>
5.Display
6.Exit
Enter the choice
3
Enter the element to be enqueued
500
[1 5*****500 100]
1.Enqueue<Queue1>
2.Dequeue<Queue1>
3.Enqueue<Queue2>
4.Dequeue<Queue2>
5.Display
6.Exit
Enter the choice
5
=====
Multiple Queues
=====
[1 5*****500 100]
1.Enqueue<Queue1>
2.Dequeue<Queue1>
3.Enqueue<Queue2>
```

```
C:\WINDOWS\system32\cmd.exe

4.Dequeue<Queue2>
5.Display
6.Exit
Enter the choice
3
Enter the element to be enqueued
100
[1 5*****100]
1.Enqueue<Queue1>
2.Dequeue<Queue1>
3.Enqueue<Queue2>
4.Dequeue<Queue2>
5.Display
6.Exit
Enter the choice
3
Enter the element to be enqueued
500
[1 5*****500 100]
1.Enqueue<Queue1>
2.Dequeue<Queue1>
3.Enqueue<Queue2>
4.Dequeue<Queue2>
5.Display
6.Exit
Enter the choice
5
=====
Multiple Queues
=====
[1 5*****500 100]
1.Enqueue<Queue1>
2.Dequeue<Queue1>
3.Enqueue<Queue2>
4.Dequeue<Queue2>
5.Display
6.Exit
Enter the choice
2
Dequeued element = 1
[5*****500 100]
1.Enqueue<Queue1>
2.Dequeue<Queue1>
3.Enqueue<Queue2>
4.Dequeue<Queue2>
5.Display
6.Exit
Enter the choice
4
Dequeued element = 100
[5*****500]
1.Enqueue<Queue1>
2.Dequeue<Queue1>
3.Enqueue<Queue2>
4.Dequeue<Queue2>
5.Display
6.Exit
Enter the choice
6

E:\nitk\semester1\dataStructures\Labpgms\Labpgm3\multipleQueues\src>
```

# Output (Lab pgm 4)

Sunday, November 5, 2017 10:18 PM

```
C:\WINDOWS\system32\cmd.exe

rtionSort.jar insertionSort
added manifest
adding: insertionSort/(in = 0) (out= 0)(stored 0%)
adding: insertionSort/InsertionSort.class(in = 2161) (out= 1180)(deflated 45%)
adding: insertionSort/InsertionSort.java(in = 1695) (out= 622)(deflated 63%)
E:\nitk\semester1\dataStructures\Labpgms\Labpgm4\src>java -jar insertionSort.jar

Enter the number of elements
5
Enter the elements
5
4
6
3
7
Pass 1: [5, 4, 6, 3, 7]
Pass 1: [5, 5, 6, 3, 7]
Pass 1: [4, 5, 6, 3, 7]

Pass 2: [4, 5, 6, 3, 7]
Pass 2: [4, 5, 6, 3, 7]

Pass 3: [4, 5, 6, 3, 7]
Pass 3: [4, 5, 6, 6, 7]
Pass 3: [4, 5, 5, 6, 7]
Pass 3: [4, 4, 5, 6, 7]
Pass 3: [3, 4, 5, 6, 7]

Pass 4: [3, 4, 5, 6, 7]
Pass 4: [3, 4, 5, 6, 7]

Sorted array = [3, 4, 5, 6, 7]
E:\nitk\semester1\dataStructures\Labpgms\Labpgm4\src>
```

```
C:\WINDOWS\system32\cmd.exe

Pass 1: [5, 5, 6, 3, 7]
Pass 1: [4, 5, 6, 3, 7]

Pass 2: [4, 5, 6, 3, 7]
Pass 2: [4, 5, 6, 3, 7]

Pass 3: [4, 5, 6, 3, 7]
Pass 3: [4, 5, 6, 6, 7]
Pass 3: [4, 5, 5, 6, 7]
Pass 3: [4, 4, 5, 6, 7]
Pass 3: [3, 4, 5, 6, 7]

Pass 4: [3, 4, 5, 6, 7]
Pass 4: [3, 4, 5, 6, 7]

Sorted array = [3, 4, 5, 6, 7]
E:\nitk\semester1\dataStructures\Labpgms\Labpgm4\src>java -jar mergeSort.jar
Enter the number of elements
5
Enter the elements
5
4
6
3
7
Unsorted array: [5, 4, 6, 3, 7]
Merged partially sorted arrays : [4, 5, 6, 3, 7]
Merged partially sorted arrays : [4, 5, 6, 3, 7]
Merged partially sorted arrays : [4, 5, 6, 3, 7]
Merged partially sorted arrays : [3, 4, 5, 6, 7]

Sorted array = [3, 4, 5, 6, 7]
E:\nitk\semester1\dataStructures\Labpgms\Labpgm4\src>
```

```
C:\WINDOWS\system32\cmd.exe

5
4
6
3
7
Unsorted array: [5, 4, 6, 3, 7]
Merged partially sorted arrays : [4, 5, 6, 3, 7]
Merged partially sorted arrays : [4, 5, 6, 3, 7]
Merged partially sorted arrays : [4, 5, 6, 3, 7]
Merged partially sorted arrays : [3, 4, 5, 6, 7]
Sorted array = [3, 4, 5, 6, 7]

E:\nitk\semester1\dataStructures\Labpgms\Labpgm4\src>java -jar heapSort.jar
Enter the number of elements
5
Enter the elements
5
4
6
3
7
Unsorted array: [5, 4, 6, 3, 7]
Max Heap = [7, 6, 5, 3, 4]
Sorting the array by getting the ith largest element at the root and adjusting the heap
Array contents =[6, 5, 4, 3, 7]
Array contents =[5, 4, 3, 6, 7]
Array contents =[4, 3, 5, 6, 7]
Array contents =[3, 4, 5, 6, 7]
Array contents =[3, 4, 5, 6, 7]
Sorted array = [3, 4, 5, 6, 7]

E:\nitk\semester1\dataStructures\Labpgms\Labpgm4\src>
```

```
C:\WINDOWS\system32\cmd.exe

Array contents =[6, 5, 4, 3, 7]
Array contents =[5, 4, 3, 6, 7]
Array contents =[4, 3, 5, 6, 7]
Array contents =[3, 4, 5, 6, 7]
Array contents =[3, 4, 5, 6, 7]
Sorted array = [3, 4, 5, 6, 7]

E:\nitk\semester1\dataStructures\Labpgms\Labpgm4\src>java -jar quickSort.jar
Enter the number of elements
5
Enter the elements
5
4
6
3
7
Unsorted array: [5, 4, 6, 3, 7]
Array = [5, 4, 3, 6, 7] i=2 j=3 pivot element = 5
Array = [5, 4, 3, 6, 7] i=3 j=2 pivot element = 5
Array = [3, 4, 5, 6, 7]
Partitioned array = [3, 4, 5, 6, 7] partition=2
Array = [3, 4, 5, 6, 7] i=1 j=0 pivot element = 3
Array = [3, 4, 5, 6, 7]
Partitioned array = [3, 4, 5, 6, 7] partition=0
Array = [3, 4, 5, 6, 7] i=4 j=3 pivot element = 6
Array = [3, 4, 5, 6, 7]
Partitioned array = [3, 4, 5, 6, 7] partition=3
Sorted array = [3, 4, 5, 6, 7]

E:\nitk\semester1\dataStructures\Labpgms\Labpgm4\src>
```

# Output (Lab pgm 5)

Sunday, November 5, 2017 10:19 PM

```
C:\WINDOWS\system32\cmd.exe
E:\nitk\semester1\dataStructures\Labpgms\Labpgm5\src>java -jar matrix.jar
Enter the dimensions of matrix 1
rows:
2
cols:
2
Enter the elements
1
3
4
5
Enter the dimensions of matrix 1
rows:
2
cols:
2
Enter the elements
1
3
4
5
=====
Matrix 1
=====
1 3
4 5
=====
Matrix 2
=====
1 3
4 5
=====
Matrix Addition
=====
2 6
8 10
=====
Matrix Subtraction
=====
0 0
0 0
=====
Matrix Multiplication
=====
13 18
24 37
E:\nitk\semester1\dataStructures\Labpgms\Labpgm5\src>
```

# Output (Lab pgm 6)

Sunday, November 5, 2017 10:20 PM

```
C:\WINDOWS\system32\cmd.exe

Enter the choice
1.Enqueue
2.Dequeue
3.Display
4.Exit
4

E:\nitk\semester1\dataStructures\Labpgms\Labpgm6\src>java -jar priorityQueue.jar

Enter the choice
1.Enqueue
2.Dequeue
3.Display
4.Exit
1
Enter the element to be inserted
10
priorityQueue = [10]
Enter the choice
1.Enqueue
2.Dequeue
3.Display
4.Exit
1
Enter the element to be inserted
50
priorityQueue = [50, 10]
Enter the choice
1.Enqueue
2.Dequeue
3.Display
4.Exit
1
Enter the element to be inserted
40
priorityQueue = [50, 10, 40]
Enter the choice
1.Enqueue
2.Dequeue
3.Display
4.Exit
2
Last element = 40
parent = 0child = 1largestChild = 1
Deleted element = 50
priorityQueue = [40, 10]
Enter the choice
1.Enqueue
2.Dequeue
3.Display
4.Exit
4

E:\nitk\semester1\dataStructures\Labpgms\Labpgm6\src>
```

# Output (Lab pgm 7)

Sunday, November 5, 2017 10:21 PM

```
C:\WINDOWS\system32\cmd.exe
E:\nitk\semester1\dataStructures\Labpgms\Labpgm7\src>java -jar binarySearchTree.jar
Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Exit
1
Enter the element to be inserted
50
=====
Binary Search Tree printed sideways
=====
50

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Exit
1
Enter the element to be inserted
25
=====
Binary Search Tree printed sideways
=====
50
  25

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Exit
1
Enter the element to be inserted
75
=====
Binary Search Tree printed sideways
=====
50
  75
    25

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Exit
1
Enter the element to be inserted
20
=====
Binary Search Tree printed sideways
=====
50
  75
    20
```



```
C:\WINDOWS\system32\cmd.exe

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Exit
1
Enter the element to be inserted
20
=====
Binary Search Tree printed sideways
=====
      75
     50
      25
       20

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Exit
1
Enter the element to be inserted
30
=====
Binary Search Tree printed sideways
=====
      75
     50
      30
      25
       20

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Exit
1
Enter the element to be inserted
70
=====
Binary Search Tree printed sideways
=====
      75
     50
      70
      30
      25
       20

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Exit
1
```

```
C:\WINDOWS\system32\cmd.exe

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Exit
1
Enter the element to be inserted
80
=====
Binary Search Tree printed sideways
=====
      80
     75
     70
     50
      30
      25
       20

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Exit
3
```

```

2.Delete
3.Search
4.Display
5.Exit
3
Enter the key to be searched
70
Key found at level 2
=====
Binary Search Tree printed sideways
=====
      80
     /  \
    75   70
   /  \
  50   30
 /  \
25  20

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Exit
2
Enter the element to be deleted
20
=====
Binary Search Tree printed sideways
=====
      80
     /  \
    75   70
   /  \
  50   30
 /  \
25   20

Enter the choice

```

```
C:\WINDOWS\system32\cmd.exe

5.Exit
3
Enter the key to be searched
70
Key found at level 2
=====
Binary Search Tree printed sideways
=====
      80
     75
    70
50   30
    25
     20

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Exit
2
Enter the element to be deleted
20
=====
Binary Search Tree printed sideways
=====
      80
     75
    70
50   30
    25

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Exit
4
=====
Binary Search Tree printed sideways
=====
      80
     75
    70
50   30
    25

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Exit
5

E:\nitk\semester1\dataStructures\Labpgms\Labpgm7\src>
```

# Output (Lab pgm 8)

Sunday, November 5, 2017 10:26 PM

```
C:\WINDOWS\system32\cmd.exe - java -jar binarySearchTree.jar

=====
Binary Search Tree printed sideways
=====
      80
     75
    70
   50
  25
 20

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Maximum
6.Minimum
7.height
8.Inorder
9.Preorder
10.Postorder
11.Level order
12.Exit
5
Max = 80

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Maximum
6.Minimum
7.height
8.Inorder
9.Preorder
10.Postorder
11.Level order
12.Exit
6
Min = 20

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Maximum
6.Minimum
7.height
8.Inorder
9.Preorder
10.Postorder
11.Level order
12.Exit
7
```

```
C:\WINDOWS\system32\cmd.exe - java -jar binarySearchTree.jar

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Maximum
6.Minimum
7.height
8.Inorder
9.Preorder
10.Postorder
11.Level order
12.Exit
7
Height of the BST = 3

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Maximum
6.Minimum
```

```

1.Insert
2.Delete
3.Search
4.Display
5.Maximum
6.Minimum
7.height
8.Inorder
9.Preorder
10.Postorder
11.Level order
12.Exit
8
=====
Binary Search Tree printed sideways
=====
      80
     75
    70
50   30
    25
     20

=====
Inorder traversal
=====
20 25 30 50 70 75 80

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Maximum
6.Minimum
7.height
8.Inorder
9.Preorder
10.Postorder
11.Level order
12.Exit
9

```

```

C:\WINDOWS\system32\cmd.exe - java -jar binarySearchTree.jar
Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Maximum
6.Minimum
7.height
8.Inorder
9.Preorder
10.Postorder
11.Level order
12.Exit
9
=====
Binary Search Tree printed sideways
=====
      80
     75
    70
50   30
    25
     20

=====
Preorder traversal
=====
50 25 20 30 75 70 80

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Maximum
6.Minimum
7.height
8.Inorder
9.Preorder
10.Postorder
11.Level order
12.Exit
10
=====
Binary Search Tree printed sideways
=====
      80
     75
    70
50   30

```

```

    80
  75  70
50   30
  25  20
=====
Postorder traversal
=====
20 30 25 70 80 75 50
Enter the choice

```

```

C:\WINDOWS\system32\cmd.exe
Binary Search Tree printed sideways
=====
    80
  75  70
50   30
  25  20
=====
Postorder traversal
=====
20 30 25 70 80 75 50

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Maximum
6.Minimum
7.height
8.Inorder
9.Preorder
10.Postorder
11.Level order
12.Exit
11
=====
Binary Search Tree printed sideways
=====
    80
  75  70
50   30
  25  20
=====
Level order traversal
=====
50 25 75 20 30 70 80

Enter the choice
1.Insert
2.Delete
3.Search
4.Display
5.Maximum
6.Minimum
7.height
8.Inorder
9.Preorder
10.Postorder
11.Level order
12.Exit
12
E:\nitk\semester1\dataStructures\Labpgms\Labpgm8\src>

```

# Output (Lab pgm 9)

Sunday, November 5, 2017 10:28 PM

```
C:\WINDOWS\system32\cmd.exe

0 File(s) 0 bytes
3 Dir(s) 243,040,296,960 bytes free

E:\nitk\semester1\dataStructures\Labpgms\Labpgm9>cd src
E:\nitk\semester1\dataStructures\Labpgms\Labpgm9\src>dir
Volume in drive E is New Volume
Volume Serial Number is 8C39-16BC

Directory of E:\nitk\semester1\dataStructures\Labpgms\Labpgm9\src

10/09/2017 05:00 PM <DIR> .
10/09/2017 05:00 PM <DIR> ..
10/09/2017 05:37 PM 48 adjacencyList.txt
10/09/2017 06:08 PM <DIR> dfs
10/09/2017 06:45 PM 7,810 dfs.jar
10/08/2017 11:12 PM 21 manifest.txt
3 File(s) 7,879 bytes
3 Dir(s) 243,040,296,960 bytes free

E:\nitk\semester1\dataStructures\Labpgms\Labpgm9\src>java -jar dfs.jar
String=0:1,2,3
String=1:4
String=2:8
String=3:6
String=4:5
String=5:6
String=6:7
String=8:9
String=
String=
=====
DFS traversal : 0 1 4 5 6 7 2 8 9 3
=====
BFS traversal : 0 1 2 3 4 8 6 5 9 7
=====

E:\nitk\semester1\dataStructures\Labpgms\Labpgm9\src>
```

# Output (Lab pgm 10)

Sunday, November 5, 2017 10:29 PM

```
C:\WINDOWS\system32\cmd.exe

Directory of E:\nitk\semester1\dataStructures\Labpgms\Labpgm10\src
10/09/2017  10:25 PM    <DIR>          .
10/09/2017  10:25 PM    <DIR>          ..
10/09/2017  10:24 PM    <DIR>          dijkstra
10/09/2017  11:29 PM                3,412 dijkstra.jar
10/09/2017  10:24 PM                31 manifest.txt
                2 File(s)                3,443 bytes
                3 Dir(s)  243,040,296,960 bytes free

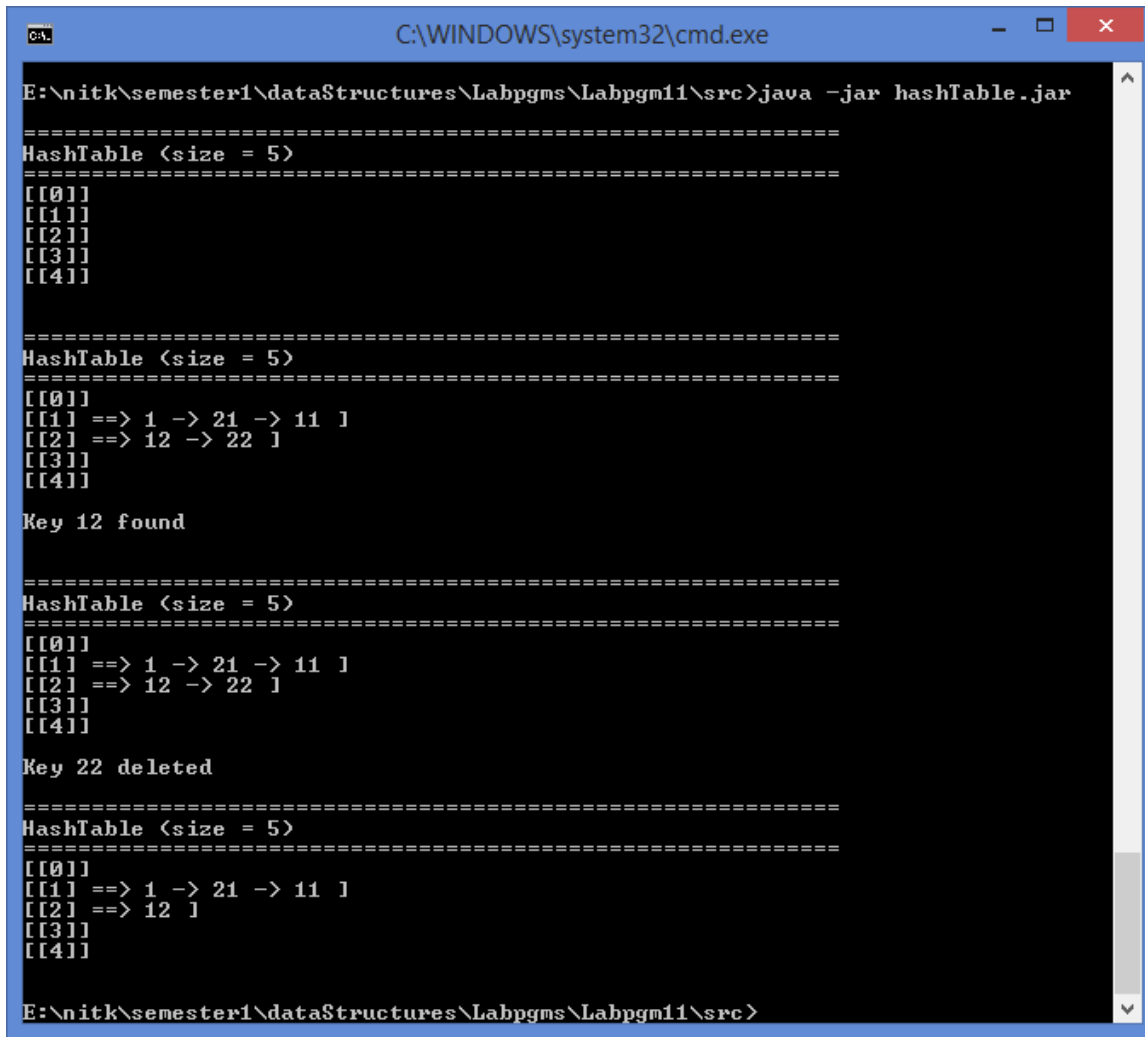
E:\nitk\semester1\dataStructures\Labpgms\Labpgm10\src>java -jar dijkstra.jar
Enter the number of nodes
3
Enter the distance matrix
0
100
50
999
0
200
999
999
0
=====
Shortest distance of all the nodes from source node 0
=====
Shortest distance of node 0 = 0 Shortest path of node 0 :
Shortest distance of node 1 = 100 Shortest path of node 1 : 0-->1
Shortest distance of node 2 = 50 Shortest path of node 2 : 0-->2

E:\nitk\semester1\dataStructures\Labpgms\Labpgm10\src>
```



# Output (Lab pgm 11)

Sunday, November 5, 2017 10:29 PM



```
C:\WINDOWS\system32\cmd.exe

E:\nitk\semester1\dataStructures\Labpgms\Labpgm11\src>java -jar hashTable.jar

=====
HashTable <size = 5>
=====
[[0]]
[[1]]
[[2]]
[[3]]
[[4]]

=====
HashTable <size = 5>
=====
[[0]]
[[1] ==> 1 -> 21 -> 11 ]
[[2] ==> 12 -> 22 ]
[[3]]
[[4]]

Key 12 found

=====
HashTable <size = 5>
=====
[[0]]
[[1] ==> 1 -> 21 -> 11 ]
[[2] ==> 12 -> 22 ]
[[3]]
[[4]]

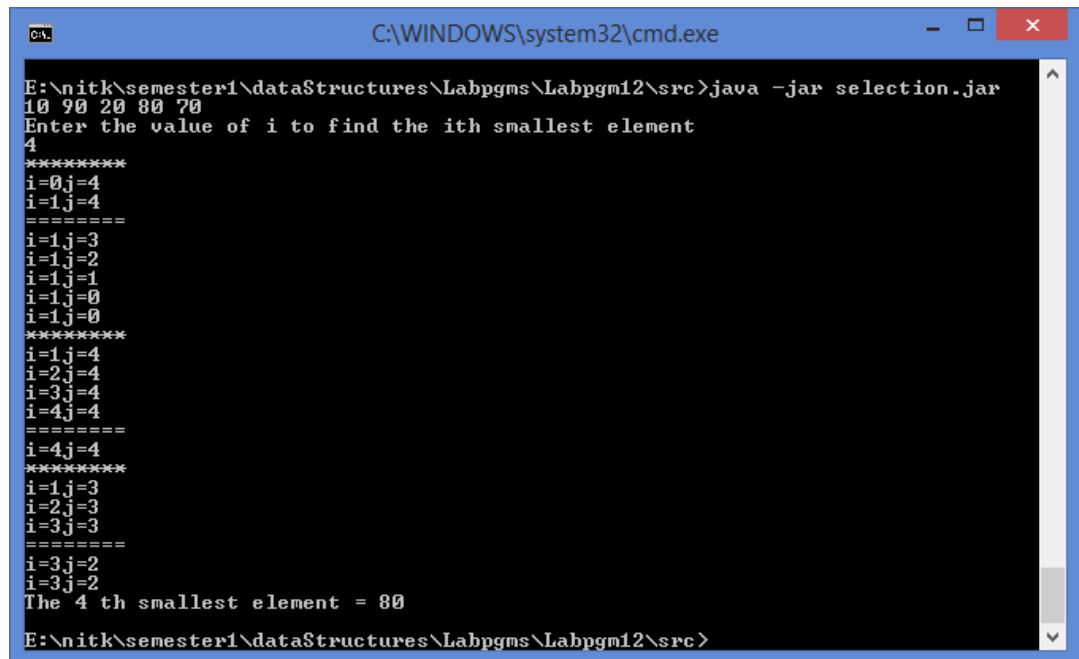
Key 22 deleted

=====
HashTable <size = 5>
=====
[[0]]
[[1] ==> 1 -> 21 -> 11 ]
[[2] ==> 12 ]
[[3]]
[[4]]

E:\nitk\semester1\dataStructures\Labpgms\Labpgm11\src>
```

# Output (Lab pgm 12)

Sunday, November 5, 2017 10:31 PM



```
C:\WINDOWS\system32\cmd.exe

E:\nitk\semester1\dataStructures\Labpgms\Labpgm12\src>java -jar selection.jar
10 90 20 80 70
Enter the value of i to find the ith smallest element
4
*****
i=0,j=4
i=1,j=4
=====
i=1,j=3
i=1,j=2
i=1,j=1
i=1,j=0
i=1,j=0
*****
i=1,j=4
i=2,j=4
i=3,j=4
i=4,j=4
=====
i=4,j=4
*****
i=1,j=3
i=2,j=3
i=3,j=3
=====
i=3,j=2
i=3,j=2
The 4 th smallest element = 80
E:\nitk\semester1\dataStructures\Labpgms\Labpgm12\src>
```

```
C:\WINDOWS\system32\cmd.exe
E:\nitk\semester1\dataStructures\Labpgms\Labpgm12\src>java -jar selectionByMedia
n.jar
10 90 20 80 70 160 150 140 130 110 100 170 180 190
Enter the value of i to find the ith smallest element
10

Inside if <n/5>>0 n=14
10 90 20 80 70 160 150 140 130 110 100 170 180 190
    count = 14

Inside while

sorted array
10 20 70 80 90
Inside while

sorted array
110 130 140 150 160
    temp array
100 170 180 190
sorted array, i=14
100 170 180 190
    count = 0

Inside if after evaluation of medians: array =
10 90 20 80 70 160 150 140 130 110 100 170 180 190
Inside if after evaluation of medians: mediansArray =
10 90 20 80 70 160 150 140 130 110 100 170 180 190
Inside if after evaluation of medians: medians =
70 140 180
Inside else <n/5>>0 n=3
70 140 180 pivot=140

*****
i=0j=13
i=1j=13pivotPosition=0
i=2j=13pivotPosition=0
i=3j=13pivotPosition=0
i=4j=13pivotPosition=0
i=5j=13pivotPosition=0
=====
i=5j=12
i=5j=11
i=5j=10
i=5j=10
swapping 160and 100

*****
i=5j=10
i=6j=10pivotPosition=0
=====
i=6j=9
i=6j=9
swapping 150and 110

*****
i=6j=9
i=7j=9pivotPosition=0
i=8j=9pivotPosition=7
i=9j=9pivotPosition=7
=====
i=9j=8
i=9j=8
swapping 140 and 130

Inside if<m>partition> m= 9 partition=8

subArray:
150 160 170 180 190
Inside if <n/5>>0 n=5
150 160 170 180 190
    count = 5

Inside while

sorted array
150 160 170 180 190
Inside if after evaluation of medians: array =
150 160 170 180 190
Inside if after evaluation of medians: mediansArray =
150 160 170 180 190
```

```
C:\WINDOWS\system32\cmd.exe

swapping 150and 110

*****
i=6j=9
i=7j=9pivotPosition=0
i=8j=9pivotPosition=7
i=9j=9pivotPosition=7
=====
i=9j=8
i=9j=8
swapping 140 and 130

Inside if<m>partition> m= 9 partition=8

subArray:
150 160 170 180 190
Inside if <n/5>>0 n=5
150 160 170 180 190
    count = 5

Inside while

sorted array
150 160 170 180 190
Inside if after evaluation of medians: array =
150 160 170 180 190
Inside if after evaluation of medians: mediansArray =
150 160 170 180 190
```

```

150 160 170 180 190
Inside if after evaluation of medians: array =
150 160 170 180 190
Inside if after evaluation of medians: mediansArray =
150 160 170 180 190
Inside if after evaluation of medians: medians =
170
Inside else <n/5>>0 n=1
170 pivot=170

*****
i=0j=4
i=1j=4pivotPosition=0
i=2j=4pivotPosition=0
i=3j=4pivotPosition=2
=====
i=3j=3
i=3j=2
i=3j=2
swapping 170 and 170

Inside if<m<partition> m= 0 partition=2

    if<m<partition> array =
150 160 170 180 190
subArray:
150 160
Inside else <n/5>>0 n=2
150 160 pivot=160

*****
i=0j=1
i=1j=1pivotPosition=0
i=2j=1pivotPosition=1
=====
i=2j=1
swapping 160 and 160

```

```

C:\WINDOWS\system32\cmd.exe
Inside if<m>partition> m= 9 partition=8

subArray:
150 160 170 180 190
Inside if <n/5>>0 n=5
150 160 170 180 190
    count = 5

Inside while

sorted array
150 160 170 180 190
Inside if after evaluation of medians: array =
150 160 170 180 190
Inside if after evaluation of medians: mediansArray =
150 160 170 180 190
Inside if after evaluation of medians: medians =
170
Inside else <n/5>>0 n=1
170 pivot=170

*****
i=0j=4
i=1j=4pivotPosition=0
i=2j=4pivotPosition=0
i=3j=4pivotPosition=2
=====
i=3j=3
i=3j=2
i=3j=2
swapping 170 and 170

Inside if<m<partition> m= 0 partition=2

    if<m<partition> array =
150 160 170 180 190
subArray:
150 160
Inside else <n/5>>0 n=2
150 160 pivot=160

*****
i=0j=1
i=1j=1pivotPosition=0
i=2j=1pivotPosition=1
=====
i=2j=1
swapping 160 and 160

Inside if<m<partition> m= 0 partition=1

    if<m<partition> array =
150 160
subArray:
150
Inside else <n/5>>0 n=1

```

```
    if(m<partition) array =  
150 160  
subArray:  
150  
Inside else (n/5)>0 n=1  
150 pivot=150  
swapping 150 and 150  
The 10 th smallest element = 150  
E:\nitk\semester1\dataStructures\Labpgms\Labpgm12\src>
```