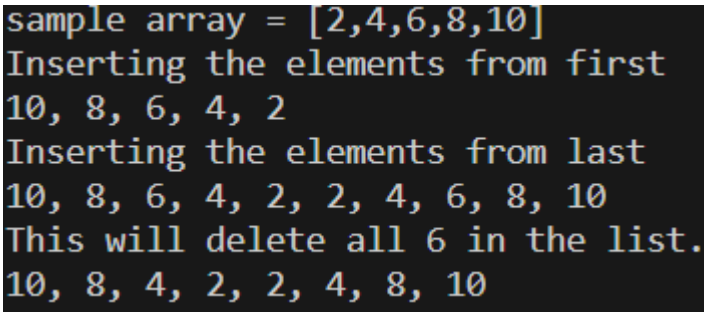
Assignment 4 Result Screenshot & brief describes.

1.

Link List:



InsertionLeft: First become new node's Next, new node become First.

InsertionRight: From First Go to last node(node -> next == nullptr), put it at nullptr.

Deletion: Remove all the given element from the list.

Searching: Check is the element in the list or not.

For Q2.

RemoveLeft: first = current ->next, and remove current.

RemoveRight: need to remember the previous node, if current -> nest == nullptr, delete the current and make previous node's next = nullptr.

2.

Using LinkList,

Stack:

Insert from left = Push

Remove from left = Pop

Queue:

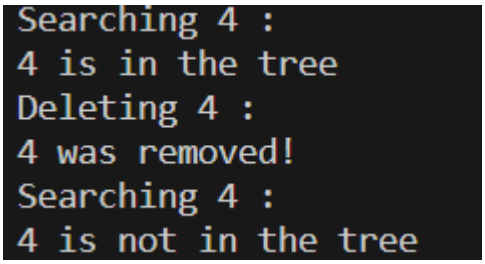
Insert from left = Enqueue

Remove from right = Dequeue

```
this is the stack
push
6
push
8, 6
push
10, 8, 6
pop
8, 6
pop
6
this is the queue
enqueue
2
enqueue
4, 2
enqueue
6, 4, 2
enqueue
8, 6, 4, 2
enqueue
10, 8, 6, 4, 2
dequeue
10, 8, 6, 4
dequeue
10, 8, 6
```

3.

Insert: 9,7,2,4,1,5 to the tree



BST tree: a<b<c : b is the parent, a is left and c is right child.

Node: data, left child, right child.

Tree: Root and Nodes.

Insert: By the rule, insert it.

Search: Search by the rule

Deleting:

1. No child: Just delete
2. One Child: Left or Right node replace the current node.
3. Two Child: The minimum node in the right sub tree become parent. By the Rule, the leftmost node in right sub tree is the closest value(and larger) to the node we are going to remove. After replacing, remove the minimum node.

4.

```
list_size: 2048


The Bubble sort operation took 4040778 times at compairing
The Bubble sort operation took 1044012 times at swaping

OperationTime: 17 ms


The Insertion sort operation took 1046052 times at compairing
The Insertion sort operation took 1044012 times at swaping

OperationTime: 4 ms


The Selection sort operation took 2096128 times at compairing
The Selection sort operation took 2048 times at swaping

OperationTime: 5 ms
Need result?(Y/N)
Y
2 3 3 4 4 6 8 8 9 10 11 12 13 13 15 15 17 17 19 19 20 21 21 21
```

```
1540 1541 1543 1547 1553 1554 1554 1556 1556 1557 1557 1558 1559 1560 1
3 1574 1574 1574 1574 1575 1577 1577 1578 1578 1578 1579 1579 1580 1580
592 1593 1595 1595 1596 1597 1598 1599 1599 1600 1604 1605 1605 1605 16
 1612 1612 1612 1612 1614 1616 1616 1616 1617 1619 1619 1620 1622 1622
32 1632 1634 1635 1636 1637 1639 1641 1642 1643 1644 1645 1645 1647 164
1655 1656 1657 1657 1659 1662 1662 1662 1663 1664 1664 1664 1665 1665 1
3 1683 1683 1684 1687 1689 1689 1691 1692 1694 1694 1694 1695 1695 1695
710 1712 1715 1715 1716 1717 1717 1719 1719 1719 1721 1722 1724 1725 17
 1731 1731 1732 1735 1736 1738 1740 1741 1741 1742 1743 1743 1745 1745
57 1757 1759 1759 1760 1761 1762 1763 1764 1765 1767 1768 1769 1772 177
1785 1786 1790 1791 1791 1792 1792 1794 1795 1796 1797 1799 1802 1803 1
9 1810 1811 1813 1813 1814 1815 1816 1817 1817 1818 1821 1822 1823 1824
830 1831 1831 1832 1833 1833 1834 1834 1836 1836 1837 1838 1839 1840 18
 1849 1851 1852 1852 1853 1853 1854 1854 1855 1856 1856 1859 1859 1859
74 1875 1875 1876 1876 1878 1879 1881 1881 1882 1883 1883 1884 1885 188
1903 1906 1909 1910 1911 1911 1912 1914 1915 1916 1916 1917 1918 1918 1
1 1933 1934 1935 1935 1935 1935 1936 1936 1938 1938 1938 1940 1940 1940
951 1952 1953 1953 1953 1954 1955 1955 1956 1957 1958 1958 1958 1959 19
 1968 1968 1970 1973 1973 1973 1974 1976 1976 1976 1976 1977 1978 1978
88 1989 1990 1990 1992 1992 1993 1993 1993 1994 1994 1997 1998 1998 199
2014 2015 2015 2015 2016 2018 2019 2020 2022 2023 2024 2025 2026 2026 2
6 2037 2038 2039 2039 2039 2040 2040 2040 2040 2042 2044 2047 2047

The results are = SAME
```

Bubble Sort: if right one is smaller, swap. Pushing the bigger one to right.
Selection Sort: Find the smallest num in the vector, put it at left. Loop the rest vector.
Insertion Sort: Keep inserting the num into an sorted vector.