

- 1) **Write a Java program to read 'N' names of your friends, store it into Hashset and display them in ascending order.**

```
import java.util.Scanner;

class SortStrings
{
    public static void main(String args[])
    {
        String temp;
        Scanner SC = new Scanner(System.in);
        System.out.print("Enter the value of N: ");
        int N= SC.nextInt();
        SC.nextLine(); //ignore next line character
        String names[] = new String[N];
        System.out.println("Enter names: ");
        for(int i=0; i<N; i++)
        {
            System.out.print("Enter name [ " + (i+1) + " ]: ");
            names[i] = SC.nextLine();
        }
        //sorting strings
        for(int i=0; i<5; i++)
        {
            for(int j=1; j<5; j++)
            {
                if(names[j-1].compareTo(names[j])>0)
                {
                    temp=names[j-1];
```

```

        names[j-1]=names[j];

        names[j]=temp;
    }

}

}

System.out.println("\nSorted names are in Ascending Order: ");

for(int i=0;i<N;i++)

{

    System.out.println(names[i]);

}

}

}

```

OUTPUT:

```

Enter the value of N: 5
Enter names:
Enter name [ 1 ]: Mike
Enter name [ 2 ]: Alex
Enter name [ 3 ]: Zoya
Enter name [ 4 ]: Yashin
Enter name [ 5 ]: Bobby

Sorted names are in Ascending Order:
Alex
Bobby
Mike
Yashin
Zoya

```

Lab Assignments

SET A

a) Write a java program to accept names of ‘n’ cities, insert same into array list collection and display the contents of same array list, also remove all these elements.

***/**

```
import java.util.*;
```

```

public class A1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        ArrayList<Object> al = new ArrayList<>();

        System.out.println("Enter How many cities :");
        int n = sc.nextInt();

        System.out.println("Enter the Cities :");
        sc.nextLine();
        for (int i = 0; i < n; i++) {
            String c = sc.nextLine();
            al.add(c);
        }
        System.out.println("Cities :" + al);

        System.out.println("ArrayList after removing the elements :");
        al.clear();

        sc.close();
    }
}

```

b) Write a java program to read ‘n’ names of your friends, store it into linked list, also display contents of the same.

```

*/
import java.util.*;
public class A2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        LinkedList<Object> ll = new LinkedList<>();

        System.out.println("Enter How many Friends :");
        int n = sc.nextInt();

        System.out.println("Enter the "+n+" Friends :");
        sc.nextLine();
        for (int i = 0; i < n; i++) {
            String fl = sc.nextLine();
            ll.add(fl);
        }
        System.out.println("Friends :" + ll);
        sc.close();
    }
}

```

c) Write a program to create a new tree set, add some colors (string) and print out the tree set.

```
import java.util.Scanner;
import java.util.TreeSet;
public class A3
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        TreeSet<Object> ts = new TreeSet<>();

        System.out.println("Enter How many Colours :");
        int n = sc.nextInt();

        System.out.println("Enter the "+n+" Colours :");
        sc.nextLine();
        for (int i = 0; i < n; i++) {
            String c = sc.nextLine();
            ts.add(c);
        }
        System.out.println("Colours :"+ ts);
        sc.close();
    }
}
```

d) Create the hash table that will maintain the mobile number and student name. Display the contact list.

```
import java.util.Hashtable;
public class A4
{
    public static void main(String[] args)
    {
        Hashtable<String, String> hashtable = new Hashtable<String, String>();
        hashtable.put("Prasad", "8796465800");
        hashtable.put("Ashish", "8806503414");
        hashtable.put("Suhas", "8629913414");
        hashtable.put("Sanket", "7118919895");
    }
}
```

```
        System.out.println(hashtable);
    } }
```

SET B

a) Accept 'n' integers from the user. Store and display integers in sorted order having proper collection class. The collection should not accept duplicate elements.

```
import java.util.TreeSet;
import java.util.Scanner;
public class B1
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        TreeSet<Object> ts = new TreeSet<>();

        System.out.println("Enter how many Numbers: ");
        int n = sc.nextInt();

        System.out.println("Enter the " + n + " Numbers: ");
        for (int i = 0; i < n; i++) {
            int num = sc.nextInt();
            ts.add(num);
        }
        System.out.println("Numbers in Sorted Order and without Duplication :"+
ts);
        sc.close();
    }
}
```

b) Write a program to sort HashMap by keys and display the details before sorting and after sorting.

```
import java.util.HashMap;
import java.util.TreeMap;

public class B2
{
    public static void main(String[] args)
    {
        HashMap<String, Integer> map = new HashMap<>();
        map.put("Prasad", 2002);
        map.put("Ashish", 2001);
        map.put("Suhas", 2002);
```

```

        map.put("Swayam", 2001);
        map.put("Sanket", 2002);
        System.out.println("\nHashMap Details Before Sorting :\n" + map);

        TreeMap<Object,Object> tm = new TreeMap<>(map);
        System.out.println("\nHashMap Details After Sorting :\n" + tm);
    }
}

```

c) Write a program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t)or(:).it takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).

```

import java.io.*;
import java.util.Hashtable;
import java.util.Scanner;
public class B3
{
    public static void main(String[] args)
    {
        Try
    {
        File f = new File("B3.txt");
        BufferedReader br = null;
        br = new BufferedReader(new FileReader(f));
        Hashtable<String, String> table = new Hashtable<>();
        Scanner sc = new Scanner(System.in);
        String line = "";
        while ((line = br.readLine()) != null) {
            String[] parts = line.split(":");

            String name = parts[0].trim();
            String number = parts[1].trim();
            if (!name.equals("") && !number.equals("")) {
                table.put(name, number);
            }
        }
        System.out.println("Enter Name :");
        String key = sc.nextLine();

        if (table.containsKey(key)) {
            System.out.println(table.get(key));
            br.close();
        }
    }
}

```

```

        sc.close();
    }
} catch (Exception e) {
    System.out.println(e);
}
}
}

```

SET C

- a) Create a java application to store city names and their STD codes using an appropriate collection. The GUI should allow the following operations:**
- i. Add a new city and its code (No duplicates)**
 - ii. Remove a city from the collection**
 - iii. Search for a city name and display the code**

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;

public class C1 extends JFrame implements ActionListener
{
    JTextField txtname, txtstd;
    JButton btnadd, btndelete, btnsearch;
    JPanel p1;
    Hashtable<String, String> table = new Hashtable<>();

    C1()
    {
        setTitle("City STD Code Information");
        setSize(700, 500);
        setVisible(true);
        setLayout(new GridLayout(3, 2, 20, 20));
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel name = new JLabel("Enter City Name: ");
        add(name);
        txtname = new JTextField(10);
        add(txtname);

        JLabel stdcode = new JLabel("Enter STD Code: ");
        add(stdcode);
        txtstd = new JTextField(10);
    }
}

```

```

add(txtstd);

JLabel op = new JLabel("Choose Operation: ");
add(op);

p1 = new JPanel();
p1.setLayout(new GridLayout(1, 3, 5, 5));

btnadd = new JButton("Add");
p1.add(btnadd);
btnadd.addActionListener(this);

btndelete = new JButton("Delete");
p1.add(btndelete);
btndelete.addActionListener(this);

btnsearch = new JButton("Search");
p1.add(btnsearch);
btnsearch.addActionListener(this);

add(p1);

} // SETC1

public void actionPerformed(ActionEvent ae) {

    String name = (txtname.getText());
    String std = (txtstd.getText());

    if (ae.getSource() == btnadd) {

        // System.out.println(table.containsKey(name) || table.containsValue(std));
        if (table.containsKey(name) || table.containsValue(std)) {
            String s2 = "Duplicates are not allowed ";
            JOptionPane.showMessageDialog(null,s2,s2,
JOptionPane.ERROR_MESSAGE);

        } else {
            table.put(name, std);
            System.out.println(table);
            JOptionPane.showMessageDialog(null, "Succesfully Added City & STD
Code", name,
JOptionPane.INFORMATION_MESSAGE);
        }
        txtname.setText("");
        txtstd.setText("");
    }
}

```



```

    }
    if (ae.getSource() == btndelete) {
        String s1 = JOptionPane.showInputDialog(null, "Enter City to remove");
        table.remove(s1);
        JOptionPane.showMessageDialog(null, "Succesfully removed City & STD
Code", name,
        JOptionPane.INFORMATION_MESSAGE);
    }

    if (ae.getSource() == btnsearch) {
        String s1 = JOptionPane.showInputDialog(null, "Enter City");

        if (table.containsKey(s1)) {
            String s2 = "STD Code: " + table.get(s1);
            JOptionPane.showMessageDialog(null, s2);
        } else {
            JOptionPane.showMessageDialog(null, "ERROR OCCURED");
        }
    }
}

}

}

public static void main(String[] args) {
    new C1();
}
}

```

b) Write a program to create link list of integer objects. Do the following:

- * i. add element at first position**
- * ii. delete last element**
- * iii. display the size of link list**

```

import java.util.LinkedList;
import java.util.Scanner;

public class C2
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        LinkedList<Object> ll = new LinkedList<>();
        ll.add(1);
        ll.add(2);
        ll.add(3);
    }
}

```

```

        System.out.println("\nElements in List :\n" + ll);

        ll.addFirst(0);

        System.out.println("\nList after adding Elements at First :\n" + ll);

        ll.removeLast();

        System.out.println("\nList after deleting Last Element :\n" + ll);

        System.out.println("\nSize of the List :\n" + ll.size());

        sc.close();
    }
}

```

c) Read a text file, specified by the first command line argument, into a list. The program should then display a menu which performs the following operations on the

list:

*** 1.Insert line 2. Delete line 3. Append line 4. Modify line 5. Exit**

When the user selects Exit, save the contents of the list to the file and end the program.

```

*/
import java.util.*;
import java.io.*;

public class C3
{
    public static void main(String arg[])
    {
        Scanner sc = new Scanner(System.in);
        try {
            File f = new File(arg[0]);
            BufferedReader br = null;

            br = new BufferedReader(new FileReader(f));
            FileOutputStream fout = new FileOutputStream(arg[0]);
            int ch;
            ArrayList<Object> al = new ArrayList<>();

            String line = "";
            while ((line = br.readLine()) != null) {
                al.add(line);
            }
        }
    }
}

```

```

    }
    do
{
    System.out.println("1.Insert Line\n2.Delete Line\n3.Append Line\n4.Modify
                        Line\n5.Exit");
    System.out.println("Enter choice: ");
    ch = sc.nextInt();
    String l1 = "This is a new Line";
    switch (ch) {
        case 1:
            al.add(l1);
            break;

        case 2:
            al.remove(l1);
            break;

        case 3:
            al.add(l1);
            break;

        case 4:
            int n = al.size() - 1;
            al.set(n, "\tUpdated line");

            break;
        case 5:
            ListIterator<Object> li = al.listIterator();
            while (li.hasNext()) {
                String l2 = (String) li.next();
                byte b[] = l2.getBytes();
                fout.write(b);

            }
            System.exit(1);
            break;

    }
    } while (ch < 6);
    br.close();
    fout.close();
} catch (Exception e) {
    System.out.println(e);
}
    sc.close();
}

```

