Collections

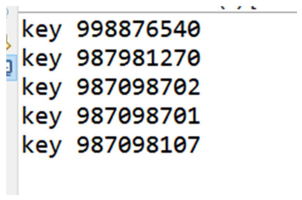
1

```
package collection;

public class Contact {
	String name;
	String email;
	Enum gender;

	public Contact(String name, String email, Enum gender) {
		super();
		this.name = name;
		this.email = email;
		this.gender = gender;
	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

	public String getEmail() {
		return email;
	}

	public void setEmail(String email) {
		this.email = email;
	}

	public Enum getGender() {
		return gender;
	}

	public void setGender(Enum gender) {
		this.gender = gender;
	}

	public String toString()
	{
		return "Contact[name= "+name+" ,email= "+email+" ,gender=
"+gender+"]";
	}

}
```

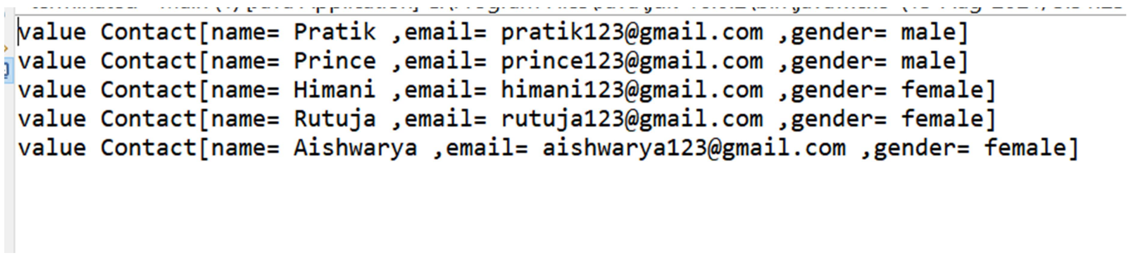
&

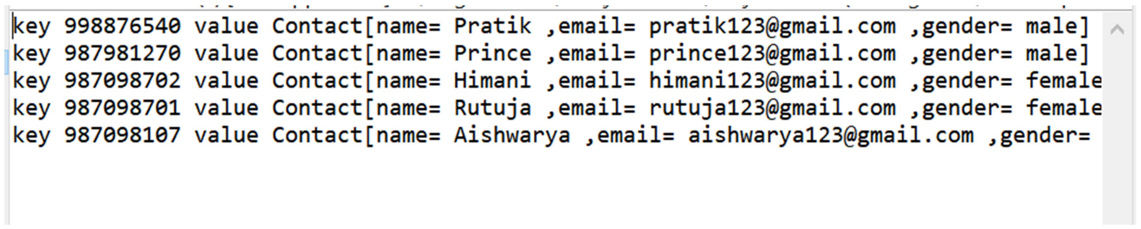
```
package collection;

import java.util.Collections;
import java.util.Iterator;
import java.util.Map;
import java.util.Set;
import java.util.TreeMap;
```

```java
public class Main {

	enum gender{
		male, female
	}

	public static void main(String[] args) {
		Main.gender gender =null;
		gender f=gender.female;
		gender m=gender.male;

		TreeMap<Long,Contact> contact= new
TreeMap<Long,Contact>(Collections.reverseOrder());

		Contact Rutuja = new Contact("Rutuja", "rutuja123@gmail.com", f);
		Contact Himani = new Contact("Himani", "himani123@gmail.com", f);
		Contact Aishwarya = new Contact("Aishwarya",
"aishwarya123@gmail.com", f);
		Contact Prince = new Contact("Prince", "prince123@gmail.com", m);
		Contact Pratik = new Contact("Pratik", "pratik123@gmail.com", m);

		contact.put((long) 987098701, Rutuja);
		contact.put((long) 987098702, Himani);
		contact.put((long) 987098107, Aishwarya);
		contact.put((long) 987981270, Prince);
		contact.put((long) 998876540, Pratik);

		Set set= contact.entrySet();
		Iterator i= set.iterator();

		while(i.hasNext())
		{
			Map.Entry save=(Map.Entry)i.next();
			System.out.println("key " +save.getKey());
			System.out.println("value " +save.getValue());
			System.out.println("key " +save.getKey()+ " value "
+save.getValue());

		}

	}

}
```



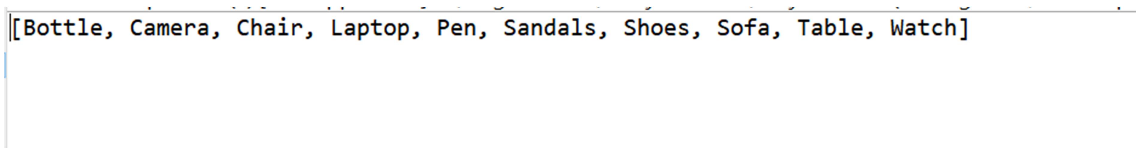
```
key 998876540
key 987981270
key 987098702
key 987098701
key 987098107
```


```
value Contact[name= Pratik ,email= pratik123@gmail.com ,gender= male]
value Contact[name= Prince ,email= prince123@gmail.com ,gender= male]
value Contact[name= Himani ,email= himani123@gmail.com ,gender= female]
value Contact[name= Rutuja ,email= rutuja123@gmail.com ,gender= female]
value Contact[name= Aishwarya ,email= aishwarya123@gmail.com ,gender= female]
```


```
key 998876540 value Contact[name= Pratik ,email= pratik123@gmail.com ,gender= male]
key 987981270 value Contact[name= Prince ,email= prince123@gmail.com ,gender= male]
key 987098702 value Contact[name= Himani ,email= himani123@gmail.com ,gender= female
key 987098701 value Contact[name= Rutuja ,email= rutuja123@gmail.com ,gender= female
key 987098107 value Contact[name= Aishwarya ,email= aishwarya123@gmail.com ,gender=
```

2

```
package collection;

import java.util.ArrayList;
import java.util.List;
import java.util.TreeSet;

public class product {

	public static void main(String[] args) {

		List<String> product = new ArrayList<String>();
		product.add("Shoes");
		product.add("Sandals");
		product.add("Watch");
		product.add("Camera");
		product.add("Laptop");
		product.add("Sofa");
		product.add("Table");
		product.add("Pen");
		product.add("Bottle");
		product.add("Chair");

		product.add("Watch");
		TreeSet<String> abc= new TreeSet<String>(product);
		System.out.println(abc);
	}

}
```


```
[Bottle, Camera, Chair, Laptop, Pen, Sandals, Shoes, Sofa, Table, Watch]
```

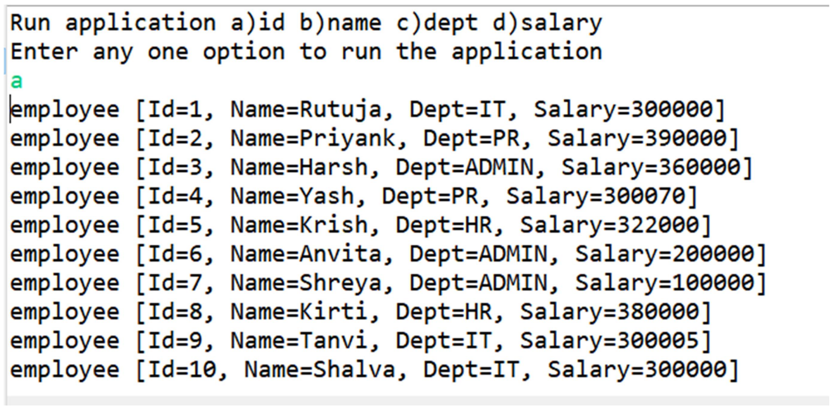
3

```
package collection;

import java.util.Objects;

public class employee implements Comparable<employee>{
	int Id;
	String Name;
	String Dept;
	int Salary;
	public employee(int id, String name, String dept, int salary) {
		super();
		this.Id = id;
		this.Name = name;
		this.Dept = dept;
		this.Salary = salary;
	}
	public int getId() {
		return Id;
	}
	public void setId(int id) {
		this.Id = id;
	}
	public String getName() {
		return Name;
	}
	public void setName(String name) {
		this.Name = name;
	}
	public String getDept() {
		return Dept;
	}
	public void setDept(String dept) {
		this.Dept = dept;
	}
	public int getSalary() {
		return Salary;
	}
	public void setSalary(int salary) {
		this.Salary = salary;
	}

	@Override
	public int hashCode() {
		return Objects.hash(Dept, Id, Name, Salary);

	}
	@Override
	public boolean equals(Object obj) {
		if (this==obj)
			return true;
		if (obj==null)
			return false;
		if (getClass()!=obj.getClass())
			return false;
		employee other= (employee) obj;
		return Objects.equals(Dept, other.Dept)&& Id==other.Id &&
Objects.equals(Name, other.Name) && Salary==other.Salary;
```

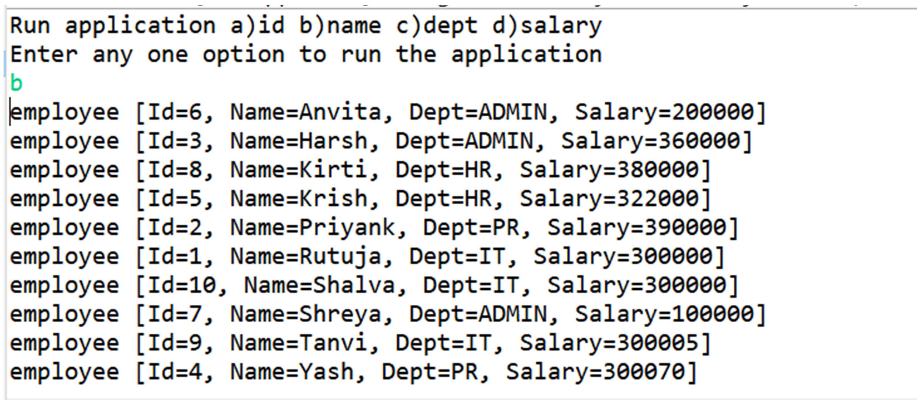
```java
	}

	@Override
	public int compareTo(employee o) {
		return this.getId()-o.getId();
	}

	@Override
	public String toString() {
		return "employee [Id=" +Id+ ", Name=" +Name+ ", Dept=" +Dept+ ", 
Salary=" +Salary+ "]";

	}

}
&

package collection;

import java.util.Iterator;
import java.util.Set;
import java.util.TreeSet;
import java.util.Scanner;
import java.util.Comparator;
import java.util.Objects;

public class sort {

	public static void main(String[] args) {
		Scanner sc= new Scanner(System.in);
		String ch;
		System.out.println("Run application a)id b)name c)dept d)salary");
		System.out.println("Enter any one option to run the application");
		ch=sc.next();

		Set<employee> set=new TreeSet<>();
		set.add(new employee(1, "Rutuja", "IT", 300000));
		set.add(new employee(8, "Kirti", "HR", 380000));
		set.add(new employee(2, "Priyank", "PR", 390000));
		set.add(new employee(4, "Yash", "PR", 300070));
		set.add(new employee(5, "Krish", "HR", 322000));
		set.add(new employee(9, "Tanvi", "IT", 300005));
		set.add(new employee(3, "Harsh", "ADMIN", 360000));
		set.add(new employee(10, "Shalva", "IT", 300000));
		set.add(new employee(6, "Anvita", "ADMIN", 200000));
		set.add(new employee(7, "Shreya", "ADMIN", 100000));


		if(ch.equals("a"))
		{
			Iterator<employee> it=set.iterator();
			while(it.hasNext())
			{
				System.out.println(it.next());
			}
		}
		else if(ch.equals("b"))
```

```
{
        set= new TreeSet<>(Comparator.comparing(employee::getName));
        set.add(new employee(1, "Rutuja", "IT", 300000));
        set.add(new employee(8, "Kirti", "HR", 380000));
        set.add(new employee(2, "Priyank", "PR", 390000));
        set.add(new employee(4, "Yash", "PR", 300070));
        set.add(new employee(5, "Krish", "HR", 322000));
        set.add(new employee(9, "Tanvi", "IT", 300005));
        set.add(new employee(3, "Harsh", "ADMIN", 360000));
        set.add(new employee(10, "Shalva", "IT", 300000));
        set.add(new employee(6, "Anvita", "ADMIN", 200000));
        set.add(new employee(7, "Shreya", "ADMIN", 100000));
        Iterator<employee> it=set.iterator();
        while(it.hasNext())
        {
                System.out.println(it.next());
        }
}
else if(ch.equals("c"))
{
        set= new TreeSet<>(Comparator.comparing(employee::getSalary));
        set.add(new employee(1, "Rutuja", "IT", 300000));
        set.add(new employee(8, "Kirti", "HR", 380000));
        set.add(new employee(2, "Priyank", "PR", 390000));
        set.add(new employee(4, "Yash", "PR", 300070));
        set.add(new employee(5, "Krish", "HR", 322000));
        set.add(new employee(9, "Tanvi", "IT", 300005));
        set.add(new employee(3, "Harsh", "ADMIN", 360000));
        set.add(new employee(10, "Shalva", "IT", 300000));
        set.add(new employee(6, "Anvita", "ADMIN", 200000));
        set.add(new employee(7, "Shreya", "ADMIN", 100000));
        Iterator<employee> it=set.iterator();
        while(it.hasNext())
        {
                System.out.println(it.next());
        }
}
else if(ch.equals("d"))
{
        set= new TreeSet<>(Comparator.comparing(employee::getDept));
        set.add(new employee(1, "Rutuja", "IT", 300000));
        set.add(new employee(8, "Kirti", "HR", 380000));
        set.add(new employee(2, "Priyank", "PR", 390000));
        set.add(new employee(4, "Yash", "PR", 300070));
        set.add(new employee(5, "Krish", "HR", 322000));
        set.add(new employee(9, "Tanvi", "IT", 300005));
        set.add(new employee(3, "Harsh", "ADMIN", 360000));
        set.add(new employee(10, "Shalva", "IT", 300000));
        set.add(new employee(6, "Anvita", "ADMIN", 200000));
        set.add(new employee(7, "Shreya", "ADMIN", 100000));
        Iterator<employee> it=set.iterator();
        while(it.hasNext())
        {
                System.out.println(it.next());
        }
}
```

```
            }
    }
```



```
Run application a)id b)name c)dept d)salary
Enter any one option to run the application
a
employee [Id=1, Name=Rutuja, Dept=IT, Salary=300000]
employee [Id=2, Name=Priyank, Dept=PR, Salary=390000]
employee [Id=3, Name=Harsh, Dept=ADMIN, Salary=360000]
employee [Id=4, Name=Yash, Dept=PR, Salary=300070]
employee [Id=5, Name=Krish, Dept=HR, Salary=322000]
employee [Id=6, Name=Anvita, Dept=ADMIN, Salary=200000]
employee [Id=7, Name=Shreya, Dept=ADMIN, Salary=100000]
employee [Id=8, Name=Kirti, Dept=HR, Salary=380000]
employee [Id=9, Name=Tanvi, Dept=IT, Salary=300005]
employee [Id=10, Name=Shalva, Dept=IT, Salary=300000]
```

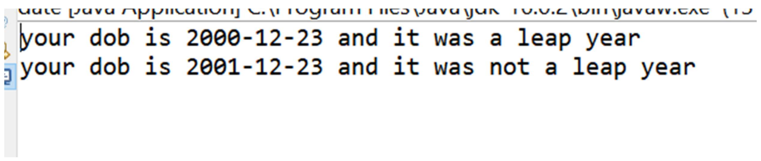


```
Run application a)id b)name c)dept d)salary
Enter any one option to run the application
b
employee [Id=6, Name=Anvita, Dept=ADMIN, Salary=200000]
employee [Id=3, Name=Harsh, Dept=ADMIN, Salary=360000]
employee [Id=8, Name=Kirti, Dept=HR, Salary=380000]
employee [Id=5, Name=Krish, Dept=HR, Salary=322000]
employee [Id=2, Name=Priyank, Dept=PR, Salary=390000]
employee [Id=1, Name=Rutuja, Dept=IT, Salary=300000]
employee [Id=10, Name=Shalva, Dept=IT, Salary=300000]
employee [Id=7, Name=Shreya, Dept=ADMIN, Salary=100000]
employee [Id=9, Name=Tanvi, Dept=IT, Salary=300005]
employee [Id=4, Name=Yash, Dept=PR, Salary=300070]
```

4

```java
package collection;

import java.time.LocalDate;
import java.util.Collection;
import java.util.Collections;
import java.util.Iterator;
import java.util.LinkedList;
```

```java
public class date {

    public static void main(String[] args) {

        LocalDate d1= LocalDate.of(2000, 12, 23);
        LocalDate d2= LocalDate.of(2001, 12, 23);

        Collection<Object> obj= new LinkedList<>();
        obj.add(d1);
        obj.add(d2);

        for (Object i:obj)
        {
            int a, c;
            int y1=d1.getYear();
            int y2=d2.getYear();

            if(y1!=0)
            {
    a=(y1%400==0)?(c=1):(y1%100==0)?(c=0):((y1%4==0)?(c=1):(c=0));
                if(a==1)
                    System.out.println("your dob is " +d1+ " and it
was a leap year");
                else
                    System.out.println("your dob is " +d1+ " and it
was not a leap year");

                if(y2!=0)
                {
    a=(y2%400==0)?(c=1):(y2%100==0)?(c=0):((y2%4==0)?(c=1):(c=0));
                    if(a==1)
                        System.out.println("your dob is " +d2+ "
and it was a leap year");
                    else
                        System.out.println("your dob is " +d2+ "
and it was not a leap year");

                    Iterator<Object> itr=obj.iterator();
                    while(itr.hasNext())
                    {
                        }
                    }

            }
        }
    }
}
```



```
date [Java Application] C:\Program Files\Java\jdk-10.0.2\bin\javaw.exe (15
your dob is 2000-12-23 and it was a leap year
your dob is 2001-12-23 and it was not a leap year
```