

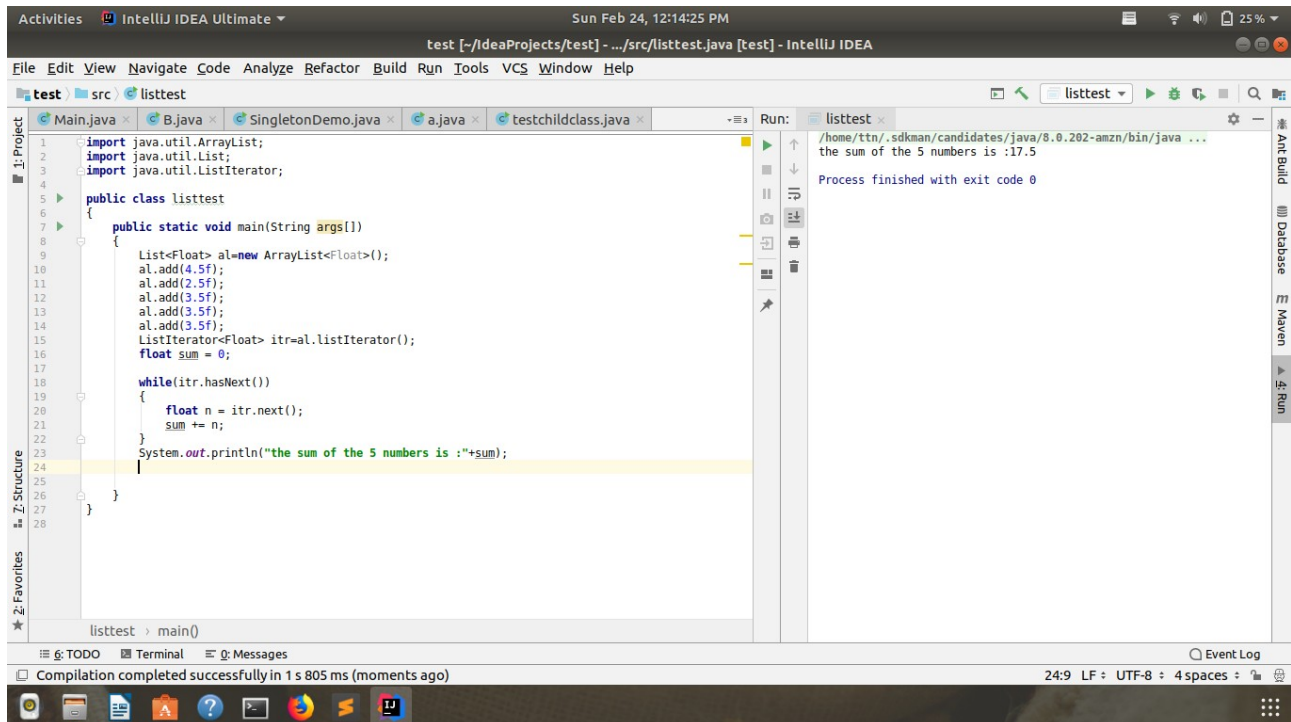
COLLECTIONS AND DATE

1. Write Java code to define List . Insert 5 floating point numbers in List, and using an iterator, find the sum of the numbers in List.
2. Write a method that takes a string and returns the number of unique characters in the string.
3. Write a method that takes a string and print the number of occurrence of each character characters in the string.
4. Write a program to sort Employee objects based on highest salary using Comparator.
Employee class{ Double Age; Double Salary; String Name
5. Write a program to sort the Student objects based on Score , if the score are same then sort on First Name . Class Student{ String Name; Double Score; Double Age
6. Print the elements of an array in the decreasing frequency if 2 numbers have same frequency then print the one which came first.
7. Design a Data Structure SpecialStack that supports all the stack operations like push(), pop(), isEmpty(), isFull() and an additional operation getMin() which should return minimum element from the SpecialStack. (Expected complexity $O(1)$)
8. Write a program to format date as example "21-March-2016"
9. Write a program to display times in different country format.

ANSWERS

Q1. Write Java code to define List . Insert 5 floating point numbers in List, and using an iterator, find the sum of the numbers in List.

ANS.



The screenshot shows the IntelliJ IDEA IDE interface. The main editor displays the code for `listtest.java`. The code imports `java.util.ArrayList`, `java.util.List`, and `java.util.ListIterator`. It defines a `listtest` class with a `main` method. In the `main` method, an `ArrayList` is created and populated with five floating-point numbers: 4.5f, 2.5f, 3.5f, 3.5f, and 3.5f. A `ListIterator` is obtained from the list, and a `while` loop iterates through the list, summing the elements. The final sum, 17.5, is printed to the console. The Run window on the right shows the output: "the sum of the 5 numbers is :17.5". The status bar at the bottom indicates that the compilation completed successfully in 1 s 805 ms.

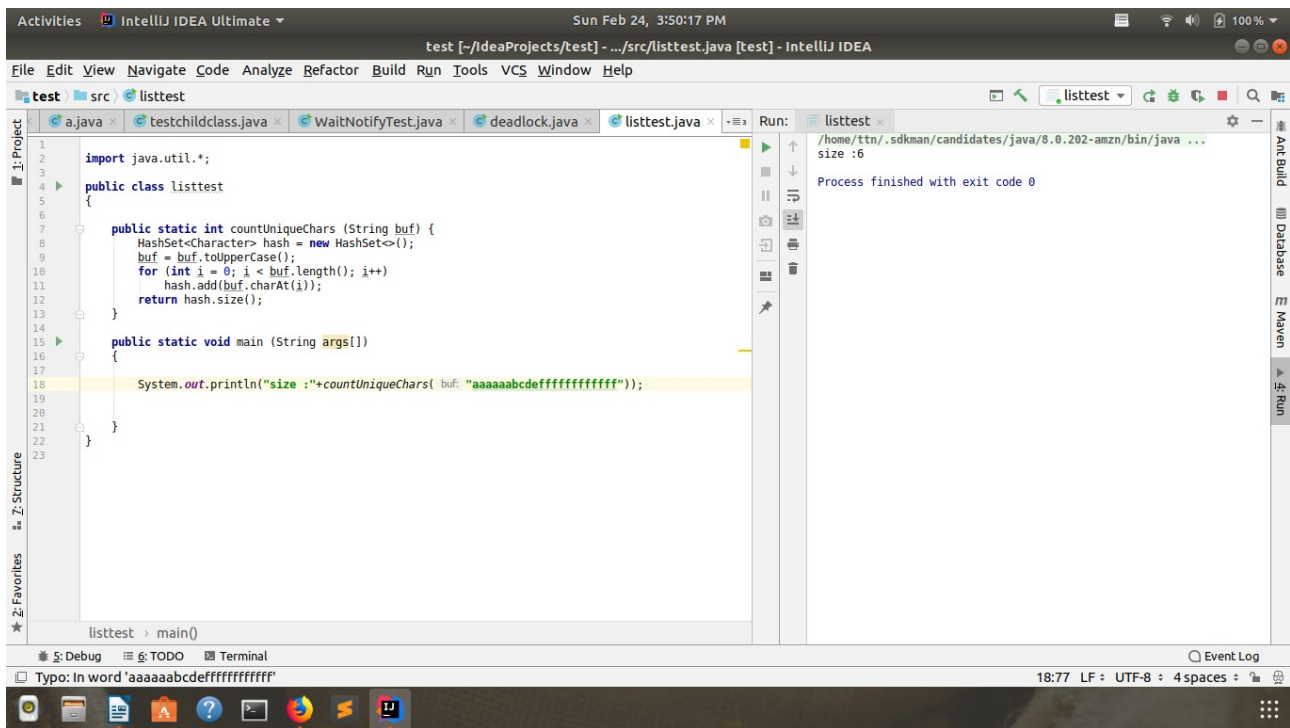
```
1 import java.util.ArrayList;
2 import java.util.List;
3 import java.util.ListIterator;
4
5 public class listtest
6 {
7     public static void main(String args[])
8     {
9         List<Float> al=new ArrayList<Float>();
10        al.add(4.5f);
11        al.add(2.5f);
12        al.add(3.5f);
13        al.add(3.5f);
14        al.add(3.5f);
15        ListIterator<Float> itr=al.listIterator();
16        float sum = 0;
17
18        while(itr.hasNext())
19        {
20            float n = itr.next();
21            sum += n;
22        }
23        System.out.println("the sum of the 5 numbers is :"+sum);
24    }
25 }
26
27
28
```

Run: listtest x
/home/ttn/.sdkman/candidates/java/8.0.202-amzn/bin/java ...
the sum of the 5 numbers is :17.5
Process finished with exit code 0

Compilation completed successfully in 1 s 805 ms (moments ago)

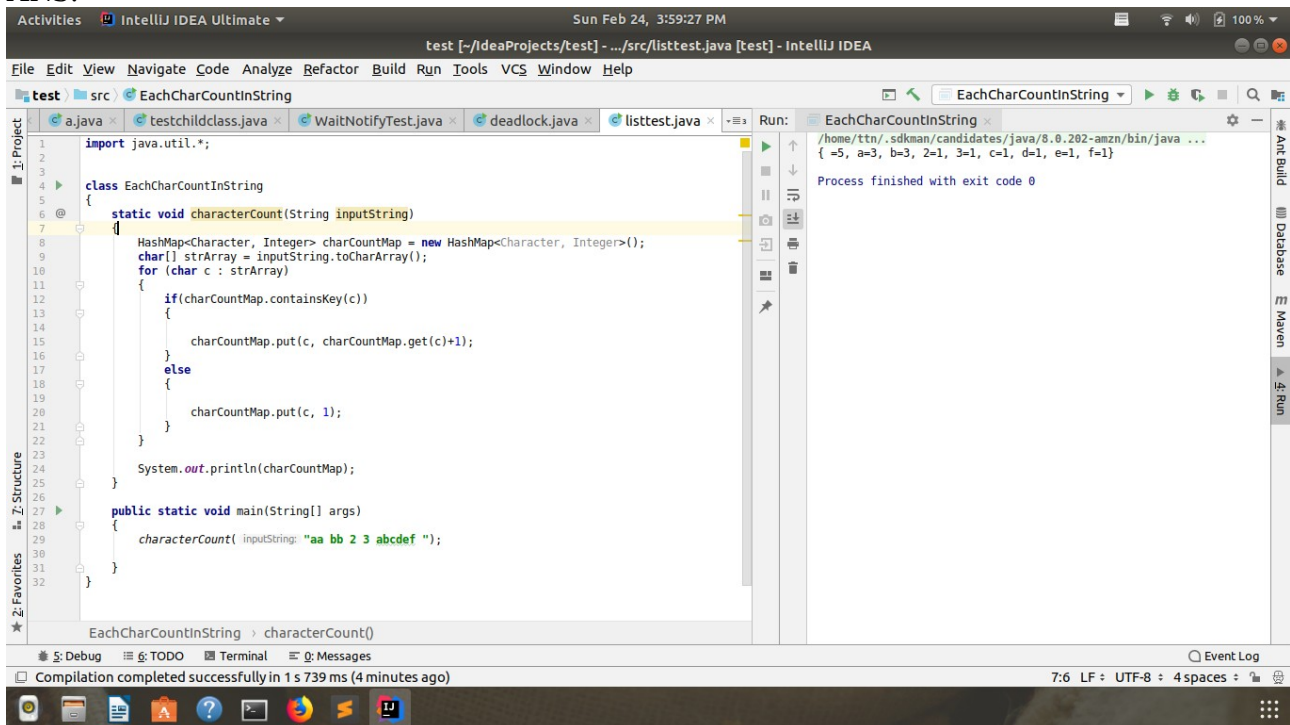
Q2. Write a method that takes a string and returns the number of unique characters in the string.

ANS.



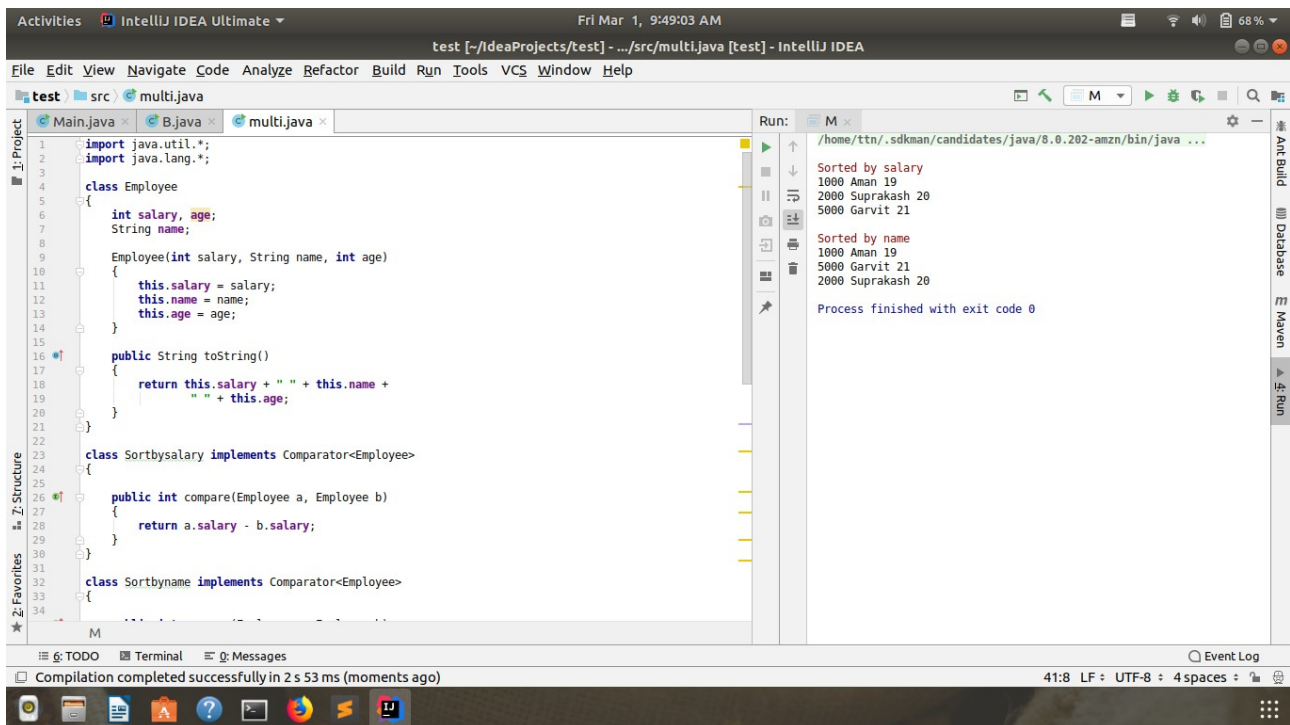
Q3. Write a method that takes a string and print the number of occurrence of each character in the string.

ANS.



Q4. Write a program to sort Employee objects based on highest salary using Comparator. Employee class{ Double Age; Double Salary; String Name

ANS.



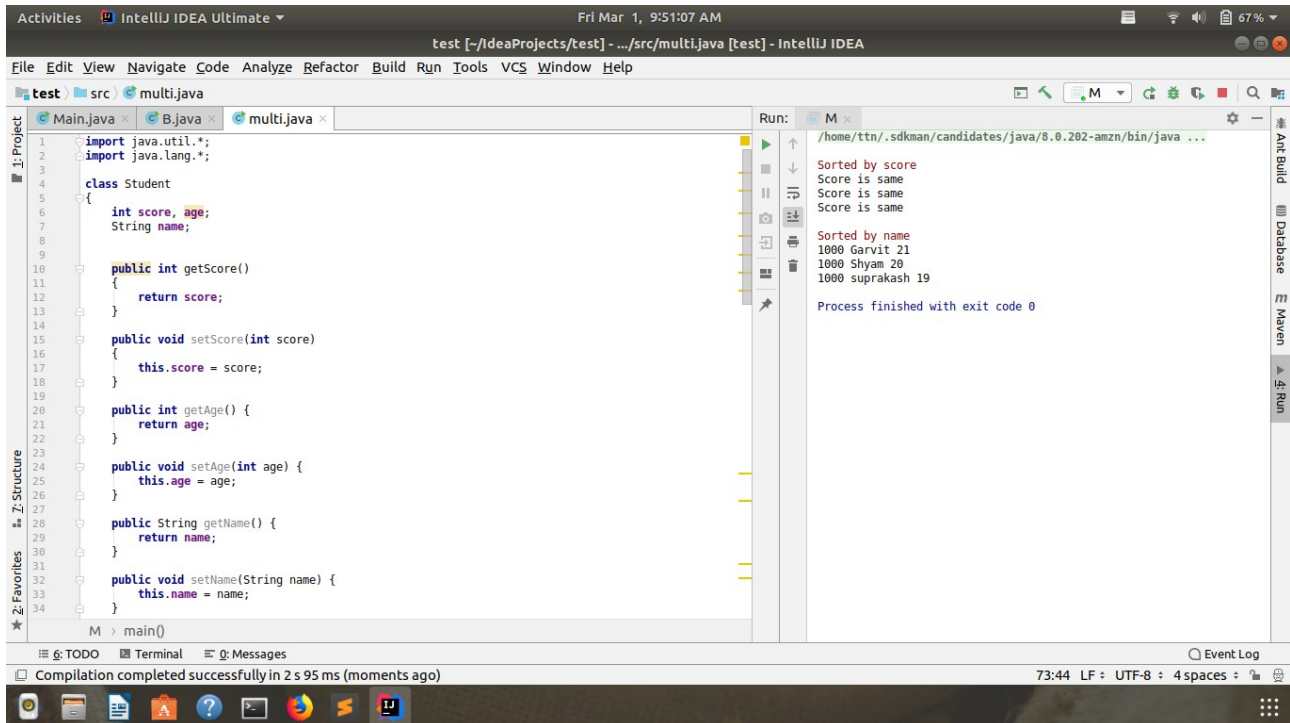
```
import java.util.*;
import java.lang.*;
class Employee
{
    int salary, age;
    String name;
    Employee(int salary, String name, int age)
    {
        this.salary = salary;
        this.name = name;
        this.age = age;
    }
    public String toString()
    {
        return this.salary + " " + this.name +
            " " + this.age;
    }
}
class Sortbysalary implements Comparator<Employee>
{
    public int compare(Employee a, Employee b)
    {
        return a.salary - b.salary;
    }
}
class Sortbyname implements Comparator<Employee>
{
    public int compare(Employee a, Employee b)
    {
        return a.name.compareTo(b.name);
    }
}
class M
{
    public static void main (String[] args)
    {
        List<Employee> ar = new ArrayList<Employee>();
        ar.add(new Employee(1000, "Aman", 19));
        ar.add(new Employee(5000, "Garvit", 21));
        ar.add(new Employee(2000, "Suprakash", 20));
    }
}
```

```

        Collections.sort(ar, new Sortbysalary());
        System.err.println("\nSorted by salary");
        for (int i=0; i<ar.size(); i++)
            System.out.println(ar.get(i));
        Collections.sort(ar, new Sortbyname());
        System.err.println("\nSorted by name");
        for (int i=0; i<ar.size(); i++)
            System.out.println(ar.get(i));
    }
}

```

Q5. Write a program to sort the Student objects based on Score, if the score are same then sort on First Name. Class Student{ String Name; Double Score; Double Age
ANS.



```

import java.util.*;
import java.lang.*;
class Student
{
    int score, age;
    String name;
    public int getScore()
    {
        return score;
    }
    public void setScore(int score)
    {
        this.score = score;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {

```

```

        this.name = name;
    }
    Student(int score, String name, int age)
    {
        this.score = score;
        this.name = name;
        this.age = age;
    }
    public String toString()
    {
        return this.score + " " + this.name +
            " " + this.age;
    }
}
class Sortbyscore implements Comparator<Student>
{
    public int compare(Student a, Student b)
    {
        return a.score - b.score;
    }
}
class Sortbyname implements Comparator<Student>
{
    public int compare(Student a, Student b)
    {
        return a.name.compareTo(b.name);
    }
}
class M
{
    public static void main (String[] args)
    {
        List<Student> ar = new ArrayList<Student>();
        ar.add(new Student(1000, "suprakash", 19));
        ar.add(new Student(1000, "Garvit", 21));
        ar.add(new Student(1000, "Shyam", 20));
        Collections.sort(ar, new Sortbyscore());
        System.err.println("\nSorted by score");
        for (int i=0; i<ar.size(); i++)
        {
            for(int j=i+1; j<ar.size(); j++)
            {
                if(ar.get(j).getScore()==ar.get(i).getScore())
                    System.out.println("Score is same");
            }
        }
        Collections.sort(ar, new Sortbyname());
        System.err.println("\nSorted by name");
        for (int i=0; i<ar.size(); i++)
            System.out.println(ar.get(i));
    }
}

```

Q6.Print the elements of an array in the decreasing frequency if 2 numbers have same frequency then print the one which came first.

ANS.


```

1 import java.util.*;
2
3 import static java.util.Map.Entry.comparingByValue;
4 import static java.util.stream.Collectors.toMap;
5 import static java.util.stream.Collectors.toSet;
6
7 class Q6 {
8
9     public static void main(String args[])
10     {
11         int arr[]={2,6,5,9,5,2,2,6,3,4,4};
12
13         Map<Integer, Integer> map = new LinkedHashMap<>();
14         for(int i=0;i<arr.length;i++)
15         {
16             if(map.containsKey(arr[i])){
17                 map.put(arr[i], map.get(arr[i]) + 1);
18             }
19             else{
20                 map.put(arr[i], 1);
21             }
22         }
23         Map<Integer, Integer> sorted = map
24             .entrySet().stream().collect(toMap(e -> e.getKey(), e -> e.getValue(), (e1, e2) -> e2,
25                 LinkedHashMap::new));
26         System.out.println("map after sorting by values: " + sorted);
27     }
28 }

```

Run: Q6

```

/home/ttn/.sdkman/candidates/java/8.0.202-amzn/bin/java ...
map after sorting by values: {2=3, 6=2, 5=2, 4=2, 9=1, 3=1}
Process finished with exit code 0

```

Q7.Design a Data Structure SpecialStack that supports all the stack operations like push(), pop(), isEmpty(), isFull() and an additional operation getMin() which should return minimum element from the SpecialStack. (Expected complexity $O(1)$)

ANS.

```

1 import java.util.Stack;
2
3 class Q7 extends Stack<Integer> {
4     Stack<Integer> minStack=new Stack<>();
5     void push(int element)
6     {
7         if(minStack.isEmpty())
8         {
9             super.push(element);
10            minStack.push(element);
11            System.out.println("pushed- "+element);
12        }
13        else
14        {
15            super.push(element);
16            int minEle=minStack.peek();
17            if(element<minEle)
18                minStack.push(element);
19            else
20                minStack.push(minEle);
21            System.out.println("pushed- "+element);
22        }
23    }
24    public Integer pop()
25    {
26        Integer poppedElement=null;
27        if(!isEmpty())
28            poppedElement=super.pop();
29        if(minStack.isEmpty()==false)
30            minStack.pop();
31        return poppedElement;
32    }
33    public Integer getMin()
34    {
35        if(isEmpty()) {

```

Run: Q7

```

/home/ttn/.sdkman/candidates/java/8.0.202-amzn/bin/java ...
pushed- 4
pushed- 3
pushed- 8
pushed- 1
pushed- 6
minimum is 1
Popped 6
Popped 1
minimum is 3
Popped 8
Popped 3
minimum is 4
Popped 4
Popped null
Process finished with exit code 0

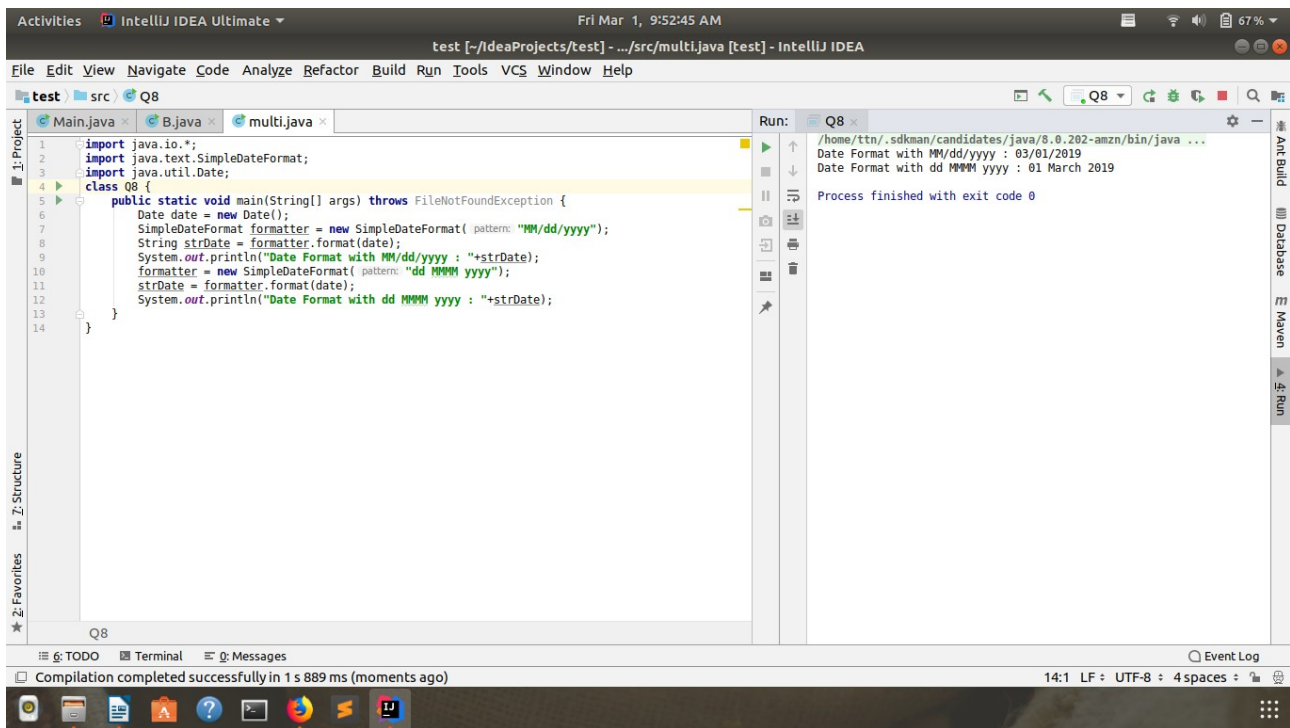
```

```

import java.util.Stack;
class Q7 extends Stack<Integer> {
    Stack<Integer> minStack=new Stack<Integer>();
    void push(int element)
    {
        if(minStack.isEmpty())
        {
            super.push(element);
            minStack.push(element);
            System.out.println("pushed- "+element);
        }
        else
        {
            super.push(element);
            int minEle=minStack.peek();
            if(element<minEle)
                minStack.push(element);
            else
                minStack.push(minEle);
            System.out.println("pushed- "+element);
        }
    }
    public Integer pop()
    {
        Integer poppedElement=null;
        if(!isEmpty())
            poppedElement=super.pop();
        if(minStack.isEmpty()==false)
            minStack.pop();
        return poppedElement;
    }
    public Integer getMin()
    {
        if(isEmpty()) {
            return null;
        }
        return minStack.peek();
    }
    public static void main(String[] args) {
        Q7 o= new Q7();
        o.push(4);
        o.push(3);
        o.push(8);
        o.push(1);
        o.push(6);
        System.out.println("\nminimum is "+ o.getMin());
        System.out.println("Popped "+ o.pop());
        System.out.println("Popped "+ o.pop());
        System.out.println("minimum is "+ o.getMin());
        System.out.println("Popped "+ o.pop());
        System.out.println("Popped "+ o.pop());
        System.out.println("minimum is "+ o.getMin());
        System.out.println("Popped "+ o.pop());
        System.out.println("Popped "+ o.pop());
    }
}

```

Q8. Write a program to format date as example "21-March-2016"
ANS.



Q9. Write a program to display times in different country format.

ANS.

