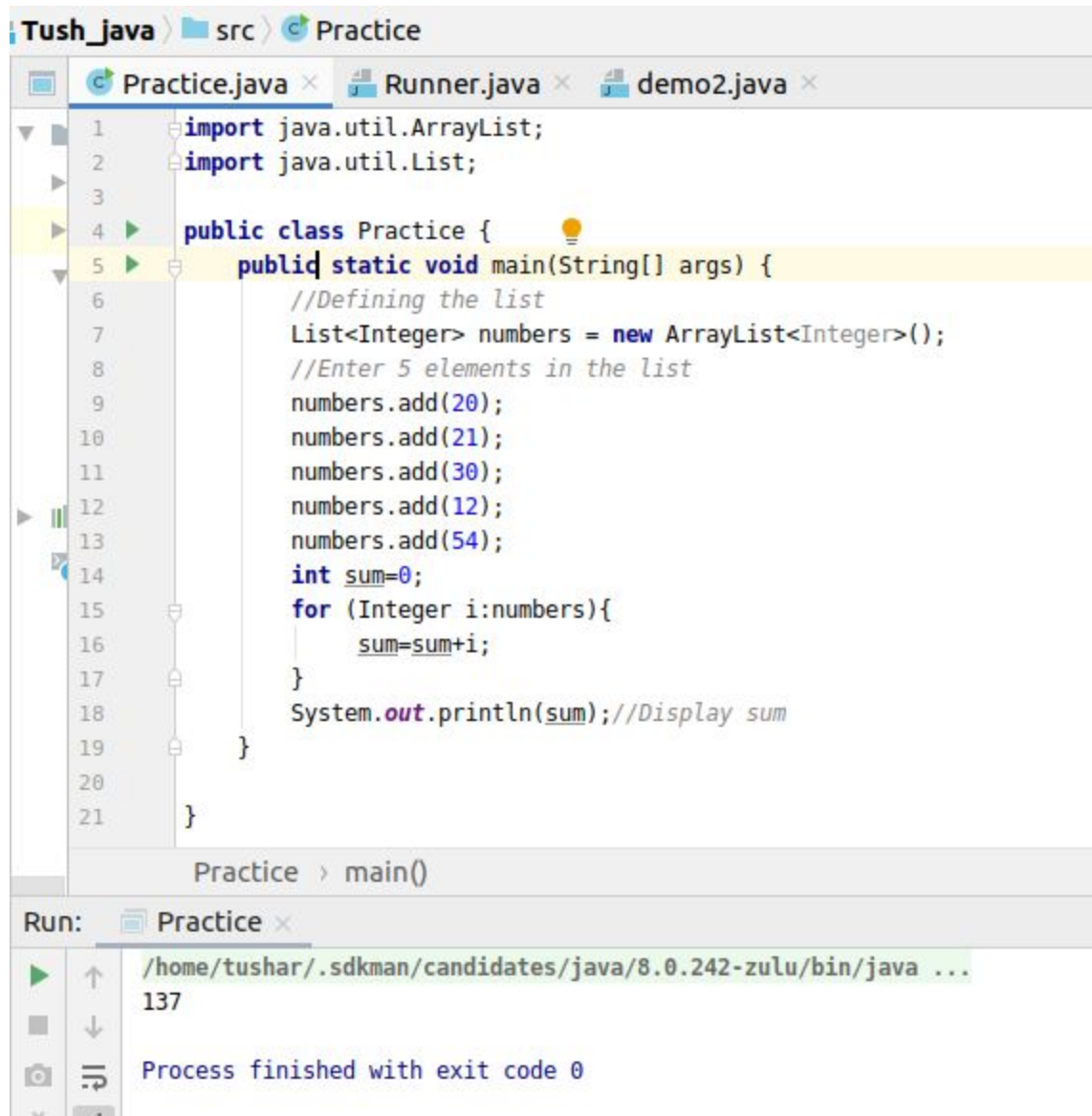


## Collections

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.

Answer:



The screenshot shows an IDE with a project named 'Tush\_java' and a source folder 'src' containing a 'Practice' package. The 'Practice.java' file is open, showing the following code:

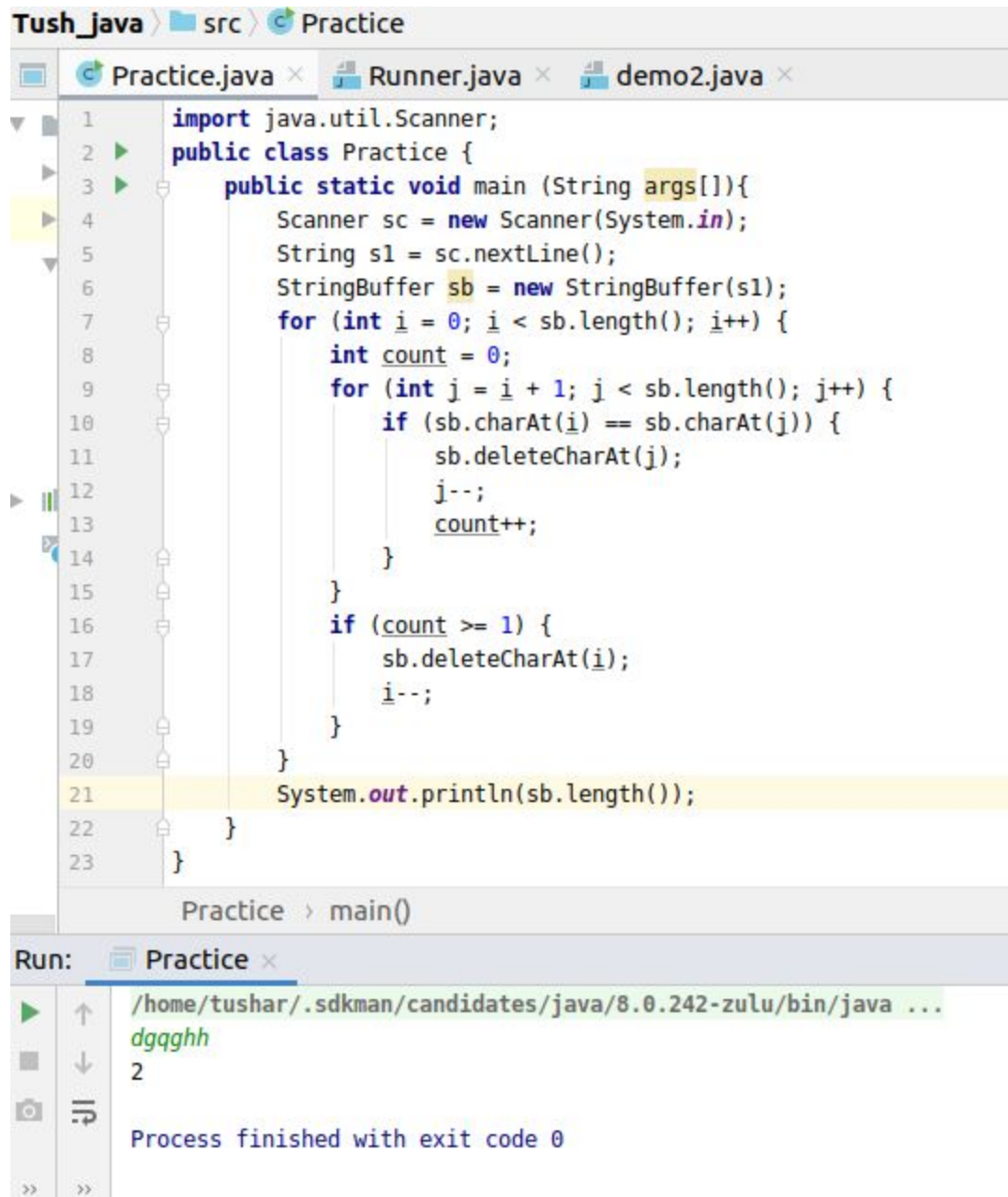
```
1 import java.util.ArrayList;
2 import java.util.List;
3
4 public class Practice {
5     public static void main(String[] args) {
6         //Defining the list
7         List<Integer> numbers = new ArrayList<Integer>();
8         //Enter 5 elements in the list
9         numbers.add(20);
10        numbers.add(21);
11        numbers.add(30);
12        numbers.add(12);
13        numbers.add(54);
14        int sum=0;
15        for (Integer i:numbers){
16            sum=sum+i;
17        }
18        System.out.println(sum); //Display sum
19    }
20 }
21 }
```

The code is executed, and the output is shown in the 'Run' console:

```
Run: Practice x
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
137
Process finished with exit code 0
```

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

Answer:



The screenshot shows an IDE with a project named 'Tush\_java' and a source folder 'src' containing a subfolder 'Practice'. The 'Practice.java' file is open, showing the following code:

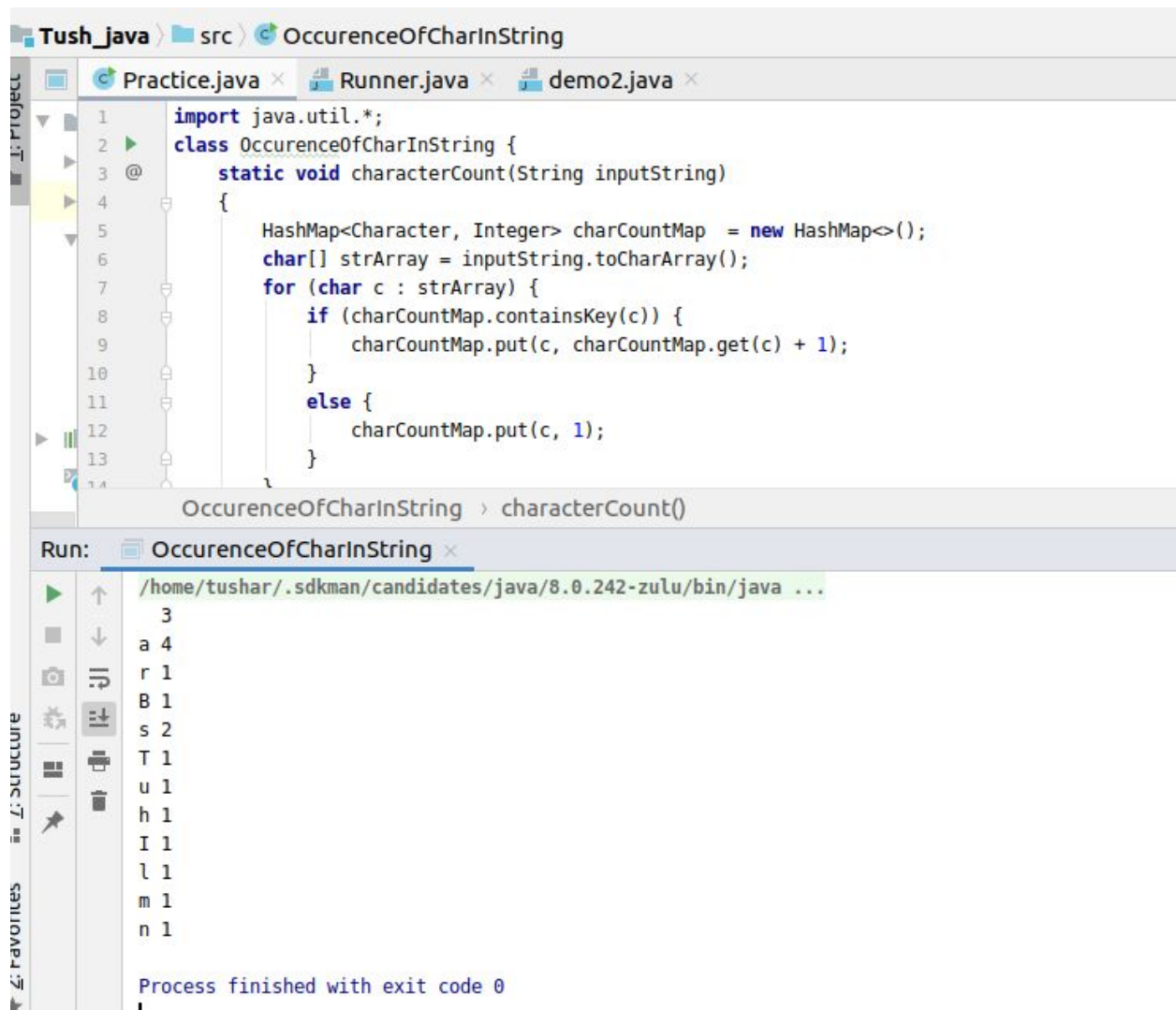
```
1  import java.util.Scanner;
2  public class Practice {
3      public static void main (String args[]){
4          Scanner sc = new Scanner(System.in);
5          String s1 = sc.nextLine();
6          StringBuffer sb = new StringBuffer(s1);
7          for (int i = 0; i < sb.length(); i++) {
8              int count = 0;
9              for (int j = i + 1; j < sb.length(); j++) {
10                 if (sb.charAt(i) == sb.charAt(j)) {
11                     sb.deleteCharAt(j);
12                     j--;
13                     count++;
14                 }
15             }
16             if (count >= 1) {
17                 sb.deleteCharAt(i);
18                 i--;
19             }
20         }
21         System.out.println(sb.length());
22     }
23 }
```

The code is executed, and the output is shown in the 'Run' window:

```
Run: Practice x
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
dgqgghh
2
Process finished with exit code 0
```

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

**Answer:**



The screenshot shows an IDE with a project named 'Tush\_java' and a source folder 'src'. The file 'OccurenceOfCharInString.java' is open, showing the following code:

```
1 import java.util.*;
2 class OccurenceOfCharInString {
3     @static void characterCount(String inputString)
4     {
5         HashMap<Character, Integer> charCountMap = new HashMap<>();
6         char[] strArray = inputString.toCharArray();
7         for (char c : strArray) {
8             if (charCountMap.containsKey(c)) {
9                 charCountMap.put(c, charCountMap.get(c) + 1);
10            }
11            else {
12                charCountMap.put(c, 1);
13            }
14        }
15    }
16 }
```

The output window shows the execution of the 'characterCount' method, displaying the frequency of each character in the string 'Practice':

```
Run: OccurenceOfCharInString x
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
3
a 4
r 1
B 1
s 2
T 1
u 1
h 1
I 1
l 1
m 1
n 1
Process finished with exit code 0
```

**Code:**

```
import java.util.*;

class OccurenceOfCharInString {

    static void characterCount(String inputString)

    {

        HashMap<Character, Integer> charCountMap = new HashMap<>();

        char[] strArray = inputString.toCharArray();

        for (char c : strArray) {
```

```

        if (charCountMap.containsKey(c)) {
            charCountMap.put(c, charCountMap.get(c) + 1);
        }
        else {
            charCountMap.put(c, 1);
        }
    }
}

for (Map.Entry entry : charCountMap.entrySet()) {
    System.out.println(entry.getKey() + " " + entry.getValue());
}
}

public static void main(String[] args)
{
    String str = "I am Tushar Bansal";
    characterCount(str);
}
}

```

**Q4. Write a program to sort HashMap by value.**

**Answer:**

**Code:**

```

import java.util.*;
import java.lang.*;

```

```

class GFG {

    public static HashMap<String, Integer> sortByValue(HashMap<String, Integer> hm)
    {
        List<Map.Entry<String, Integer> > list = new LinkedList<Map.Entry<String, Integer> >(hm.entrySet());

        Collections.sort(list, new Comparator<Map.Entry<String, Integer> >() {

            public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2){

                return (o1.getValue()).compareTo(o2.getValue());

            }

        });

        HashMap<String, Integer> temp = new LinkedHashMap<>();

        for (Map.Entry<String, Integer> aa : list) {

            temp.put(aa.getKey(), aa.getValue());

        }

        return temp;

    }

    public static void main(String[] args)
    {

        HashMap<String, Integer> hm = new HashMap<>();

        hm.put("Computer Science", 95);

        hm.put("Data Structure", 88);

        hm.put("Introduction to Database", 81);

        hm.put("Java", 85);

        hm.put("Operating Systems", 91);

        hm.put("Networking", 80);

        Map<String, Integer> hm1 = sortByValue(hm);

        for (Map.Entry<String, Integer> en : hm1.entrySet()) {

            System.out.println("Key = " + en.getKey() + ", Value = " + en.getValue());

        }

    }

}

```

```
Tush_java > src > GFG
Practice.java x Runner.java x demo2.java x
1 import java.util.*;
2 import java.lang.*;
3 class GFG {
4     public static HashMap<String, Integer> sortByValue(HashMap<String, Integer> hm)
5     {
6         List<Map.Entry<String, Integer> > list = new LinkedList<Map.Entry<String, Integer> >(hm.entrySet());
7         Collections.sort(list, new Comparator<Map.Entry<String, Integer> >() {
8             public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2){
9                 return (o1.getValue()).compareTo(o2.getValue());
10            }
11        });
12        HashMap<String, Integer> temp = new LinkedHashMap<>();
13        for (Map.Entry<String, Integer> aa : list) {
14            temp.put(aa.getKey(), aa.getValue());
15        }
16        return temp;
17    }
18    public static void main(String[] args)
19    {
20        HashMap<String, Integer> hm = new HashMap<>();
21        hm.put("Computer Science", 95);
22        hm.put("Data Structure", 88);
23    }
24}
GFG > sortByValue()

Run: GFG x
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Key = Networking, Value = 80
Key = Introduction to Database, Value = 81
Key = Java, Value = 85
Key = Data Structure, Value = 88
Key = Operating Systems, Value = 91
Key = Computer Science, Value = 95
```

**Q5. Write a program to sort Employee objects based on highest salary using Comparator. Employee class{ Double Age; Double Salary; String Name}**

**Answer:**

**Code:**

```
import java.util.*;

class Employee{

    String name;
```

```

int id;

int salary;

public Employee(int salary,String name,int id){

    this.id=id;

    this.name=name;

    this.salary=salary;

}

}

class main{

    public static void main(String[] args) {

        Employee e1 = new Employee(35000,"Tushar", 4156);

        Employee e2 = new Employee(50000,"Yadav", 4186 );

        Employee e3 = new Employee(40000,"ravi", 4158 );

        List<Employee> employees = new ArrayList<Employee>();

        employees.add(e2);

        employees.add(e3);

        employees.add(e1);

        System.out.println(employees);

        Collections.sort(employees, new Comparator<Employee>() {

            @Override

            public int compare(Employee o1, Employee o2) {

                return o1.salary - o2.salary;

            }

        });

        for (Employee i: employees)

            System.out.println("id: "+i.id+" ,name: "+i.name+" ,salary:"+i.salary);

    }

}

```



Tush\_java > src > demo2.java

Practice.java x Runner.java x demo2.java x

```
1  import java.util.*;
2  class Employee{
3      String name;
4      int id;
5      int salary;
6      public Employee(int salary,String name,int id){
7          this.id=id;
8          this.name=name;
9          this.salary=salary;
10     }
11 }
12 class main{
13     public static void main(String[] args) {
14         Employee e1 = new Employee( salary: 35000, name: "Tushar", id: 4156);
15         Employee e2 = new Employee( salary: 50000, name: "Yadav", id: 4186 );
16         Employee e3 = new Employee( salary: 40000, name: "ravi", id: 4158 );
17         List<Employee> employees = new ArrayList<Employee>();
18         employees.add(e2);
19         employees.add(e3);
20         employees.add(e1);
21         System.out.println(employees);
22         Collections.sort(employees, new Comparator<Employee>() {
```

main > main()

Run: main x

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
[Employee@7d4991ad, Employee@28d93b30, Employee@1b6d3586]
id: 4156 ,name: Tushar ,salary:35000
id: 4158 ,name: ravi ,salary:40000
id: 4186 ,name: Yadav ,salary:50000

Process finished with exit code 0
```

4: Run 6: TODO Terminal 0: Messages



**Q6.**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}.

**Answer:**

**Code:**

```
import java.util.*;
class Employee{
    String name;
    int score;
    int age;
    public Employee(int age,String name,int score){
        this.score=score;
        this.name=name;
        this.age=age;
    }
}
class main{
    public static void main(String[] args) {
        Employee e1 = new Employee(20,"Tushar", 4156);
        Employee e2 = new Employee(50,"Yadav", 4186 );
        Employee e3 = new Employee(40,"ravi", 4158 );
        Employee e4 = new Employee(30,"Dixit", 4158 );
        List<Employee> employees = new ArrayList<Employee>();
        employees.add(e2);
        employees.add(e3);
        employees.add(e1);
        System.out.println(employees);
        Collections.sort(employees, new Comparator<Employee>() {
            @Override
            public int compare(Employee o1, Employee o2) {
                return o1.score - o2.score;
            }
        });
        for (Employee i: employees)
            System.out.println("id: "+i.age+" ,name: "+i.name+" ,salary:"+i.score);
    }
}
```

```
Tushar_java > src > demo2.java
Practice.java x Runner.java x demo2.java x
1  import java.util.*;
2  class Employee{
3      String name;
4      int score;
5      int age;
6      public Employee(int age,String name,int score){
7          this.score=score;
8          this.name=name;
9          this.age=age;
10     }
11 }
12 class main{
13     public static void main(String[] args) {
14         Employee e1 = new Employee( age: 20, name: "Tushar", score: 4156);
15         Employee e2 = new Employee( age: 50, name: "Yadav", score: 4186 );
16         Employee e3 = new Employee( age: 40, name: "ravi", score: 4158 );
17         Employee e4 = new Employee( age: 30, name: "Dixit", score: 4158 );
18         List<Employee> employees = new ArrayList<Employee>();
19         employees.add(e2);
20         employees.add(e3);
21         employees.add(e1);
22         System.out.println(employees);
    }
}
main > main()

Run: main x
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
[Employee@7d4991ad, Employee@28d93b30, Employee@1b6d3586]
id: 20 ,name: Tushar ,salary:4156
id: 40 ,name: ravi ,salary:4158
id: 50 ,name: Yadav ,salary:4186
Process finished with exit code 0
```

**Q7.**Print the elements of an array in the decreasing frequency if 2 numbers have same frequency then print the one which came first.

Answer:

Code:

```
import java.io.IOException;
import java.util.*;
class demo {
    public static StringBuffer sortByFrequency(int arr[], int l1) {
        Map<Integer, Integer> countMap = getCountMap(arr1, l1);
        StringBuffer result = new StringBuffer();
        countMap.entrySet().stream()
            .sorted(Map.Entry.<Integer, Integer> comparingByValue().reversed())
            .forEach(e -> {
                int key = e.getKey();
                int val = e.getValue();
                for (int i = 0; i < val; i++) {
                    result.append(key + " ");
                }
            });
        return result;
    }
    private static Map<Integer, Integer> getCountMap(int[] arr, int l1) {
        Map<Integer, Integer> countMap = new LinkedHashMap<>();
        for (int i = 0; i < l1; i++) {
            if (countMap.containsKey(arr[i])) {
                countMap.put(arr[i], countMap.get(arr[i]) + 1);
            } else {
                countMap.put(arr[i], 1);
            }
        }
        return countMap;
    }
    public static void main(String[] args) throws IOException {
        int a[] = { 1,21, 5, 2, 6, -1, 99, 5, 8, 8,21, 8 };
        System.out.println(sortByFrequency(a, a.length));
    }
}
```

The screenshot shows an IDE with a project named 'Tush\_java' and a source folder 'src' containing a sub-folder 'demo'. Three files are open: 'Practice.java', 'Runner.java', and 'demo2.java'. The 'demo2.java' file contains the following code:

```
18  @private static Map<Integer, Integer> getCountMap(int[] arr, int l1) {
19      Map<Integer, Integer> countMap = new LinkedHashMap<>();
20      for (int i = 0; i < l1; i++) {
21          if (countMap.containsKey(arr[i])) {
22              countMap.put(arr[i], countMap.get(arr[i]) + 1);
23          } else {
24              countMap.put(arr[i], 1);
25          }
26      }
27      return countMap;
28  }
29  public static void main(String[] args) throws IOException {
30      int a[] = { 1,21, 5, 2, 6, -1, 99, 5, 8, 8,21, 8 };
31      System.out.println(sortByFrequency(a, a.length));
32  }
33  }
34
```

Below the code editor, there is a breadcrumb trail: 'demo > sortByFrequency() > e->{...}'. Below that is a 'Run:' section with a tab for 'demo'. It shows the command executed: '/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...' and the output: '8 8 8 21 21 5 5 1 2 6 -1 99'. At the bottom, it states 'Process finished with exit code 0'.

**Q8.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.**

**Answer:**

**Code:**

```
import java.util.*;
class MyStack
```

```

{
    Stack<Integer> s;
    Integer minEle;
    MyStack() { s = new Stack<Integer>(); }
    void getMin()
    {
        if (s.isEmpty())
            System.out.println("Stack is empty");
        else
            System.out.println("Minimum Element in the " + " stack is: " + minEle);
    }
    void peek()
    {
        if (s.isEmpty())
        {
            System.out.println("Stack is empty ");
            return;
        }

        Integer t = s.peek(); // Top element.
        System.out.print("Top Most Element is: ");

        if (t < minEle)
            System.out.println(minEle);
        else
            System.out.println(t);
    }
    void pop()
    {
        if (s.isEmpty())
        {
            System.out.println("Stack is empty");
            return;
        }

        System.out.print("Top Most Element Removed: ");
        Integer t = s.pop();
        if (t < minEle)
        {
            System.out.println(minEle);
            minEle = 2*minEle - t;
        }

        else
            System.out.println(t);
    }
    void push(Integer x)
    {
        if (s.isEmpty())
        {

```

```

        minEle = x;
        s.push(x);
        System.out.println("Number Inserted: " + x);
        return;
    }
    if (x < minEle)
    {
        s.push(2*x - minEle);
        minEle = x;
    }

    else
        s.push(x);

    System.out.println("Number Inserted: " + x);
}
};
class Main
{
    public static void main(String[] args)
    {
        MyStack s = new MyStack();
        s.push(3);
        s.push(5);
        s.getMin();
        s.push(2);
        s.push(1);
        s.getMin();
        s.pop();
        s.getMin();
        s.pop();
        s.peek();
    }
}

```

Tush\_java > src > demo2.java

Practice.java × Runner.java × demo2.java ×

```
69     };
70     class Main
71     {
72     public static void main(String[] args)
73     {
74         MyStack s = new MyStack();
75         s.push( x: 3);
76         s.push( x: 5);
77         s.getMin();
78         s.push( x: 2);
79         s.push( x: 1);
80         s.getMin();
81         s.pop();
82         s.getMin();
83         s.pop();
84         s.peek();
85     }
86 }
```

Run: Main ×

```
/home/tushar/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Number Inserted: 3
Number Inserted: 5
Minimum Element in the stack is: 3
Number Inserted: 2
Number Inserted: 1
Minimum Element in the stack is: 1
Top Most Element Removed: 1
Minimum Element in the stack is: 2
Top Most Element Removed: 2
Top Most Element is: 5
```

4: Run 6: TODO Terminal