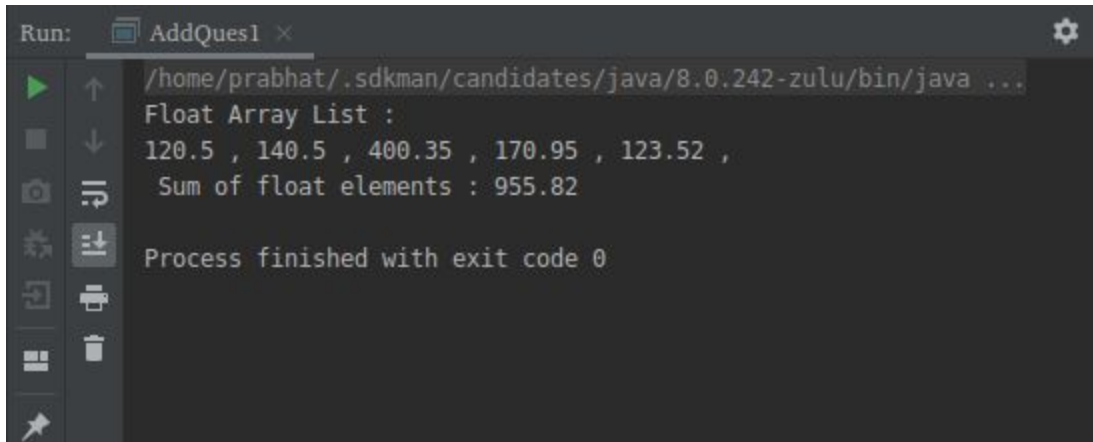


1. *Write Java code to define List . Insert 5 floating point numbers in List, and using an iterator, find the sum of the numbers in List.

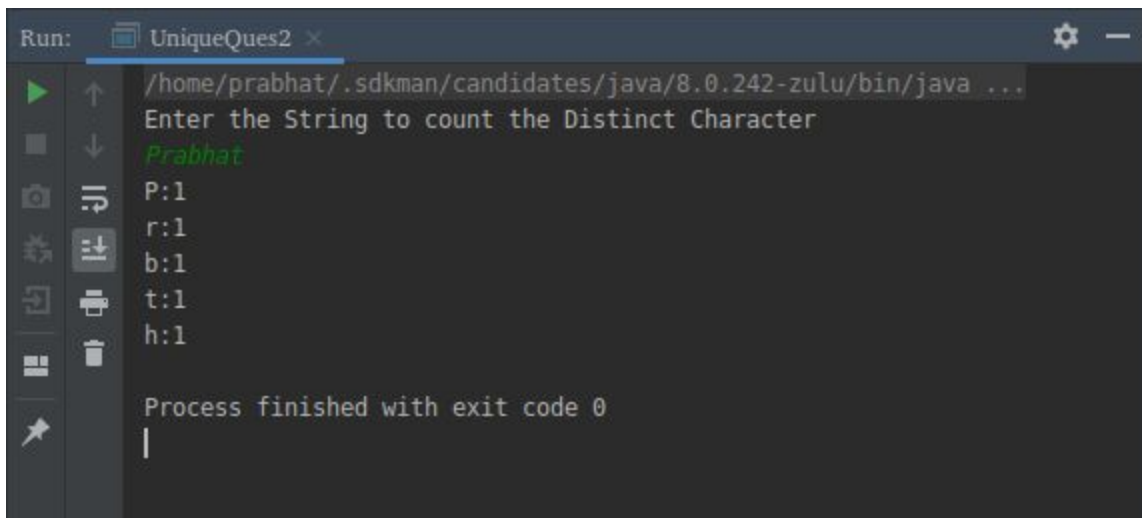
OUTPUT (AddQues1.java)



```
Run: AddQues1 x
/home/prabhat/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Float Array List :
120.5 , 140.5 , 400.35 , 170.95 , 123.52 ,
Sum of float elements : 955.82
Process finished with exit code 0
```

-
2. *Write a method that takes a string and returns the number of unique characters in the string.

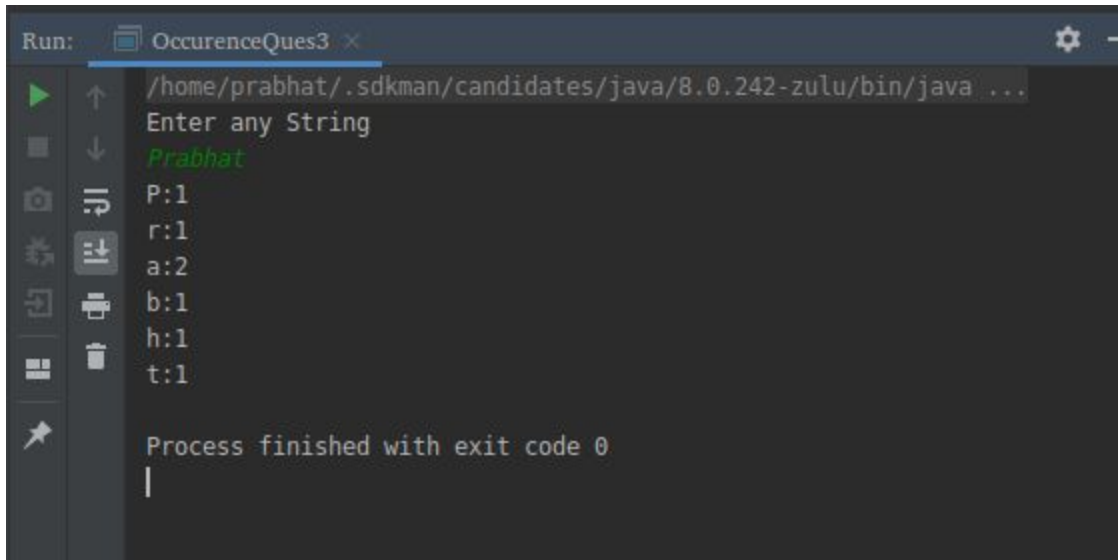
OUTPUT (UniqueQues2.java)



```
Run: UniqueQues2 x
/home/prabhat/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Enter the String to count the Distinct Character
Prabhat
P:1
r:1
b:1
t:1
h:1
Process finished with exit code 0
|
```

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

OUTPUT(OccurenceQues3.java)

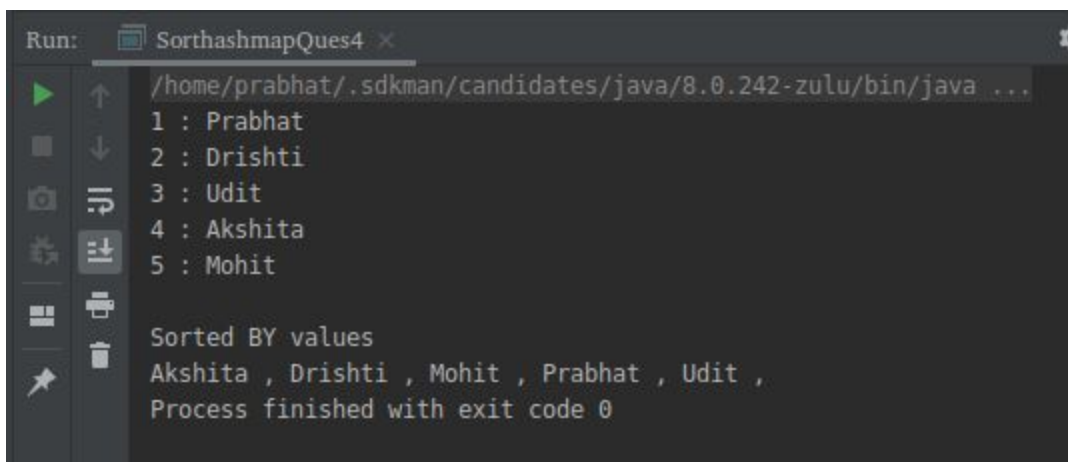


```
Run: OccurenceQues3 x
/home/prabhat/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Enter any String
Prabhat
P:1
r:1
a:2
b:1
h:1
t:1

Process finished with exit code 0
```

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

OUTPUT(SortHashmapQues4.java)

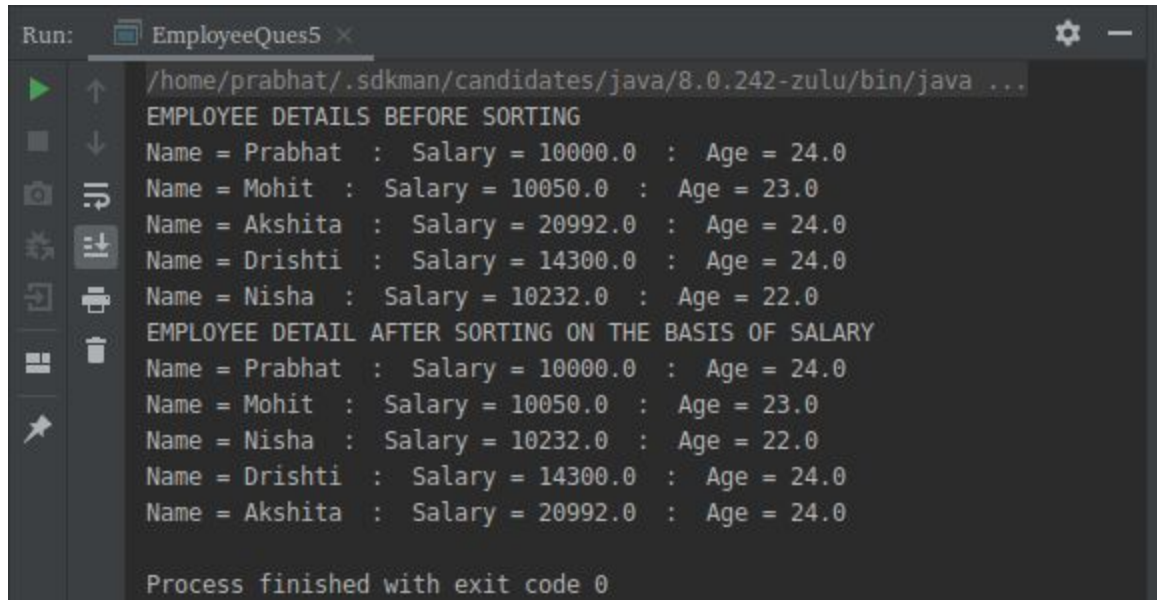


```
Run: SorthashmapQues4 x
/home/prabhat/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
1 : Prabhat
2 : Drishti
3 : Udit
4 : Akshita
5 : Mohit

Sorted BY values
Akshita , Drishti , Mohit , Prabhat , Udit ,
Process finished with exit code 0
```

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

OUTPUT(EmployeeQues5.java)



```
Run: EmployeeQues5 x
/home/prabhat/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
EMPLOYEE DETAILS BEFORE SORTING
Name = Prabhat : Salary = 10000.0 : Age = 24.0
Name = Mohit : Salary = 10050.0 : Age = 23.0
Name = Akshita : Salary = 20992.0 : Age = 24.0
Name = Drishti : Salary = 14300.0 : Age = 24.0
Name = Nisha : Salary = 10232.0 : Age = 22.0
EMPLOYEE DETAIL AFTER SORTING ON THE BASIS OF SALARY
Name = Prabhat : Salary = 10000.0 : Age = 24.0
Name = Mohit : Salary = 10050.0 : Age = 23.0
Name = Nisha : Salary = 10232.0 : Age = 22.0
Name = Drishti : Salary = 14300.0 : Age = 24.0
Name = Akshita : Salary = 20992.0 : Age = 24.0
Process finished with exit code 0
```

-
6. *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

OUTPUT(StudentQues6.java)

```
Run: StudentQus6 x
/home/prabhat/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
DATA BEFORE SORTING
Student Details --- Student Name = Prabbhat : Score = 100.0 : Age = 23.0
Student Details --- Student Name = Jerry : Score = 88.08 : Age = 24.0
Student Details --- Student Name = Zen : Score = 88.76 : Age = 24.0
Student Details --- Student Name = Tom : Score = 33.23 : Age = 21.0
Student Details --- Student Name = Akshita : Score = 98.09 : Age = 22.0
Student Details --- Student Name = Udit : Score = 77.45 : Age = 21.0
Student Details --- Student Name = Shreya : Score = 80.76 : Age = 23.0

DATA AFTER SORTING
Student Details --- Student Name = Tom : Score = 33.23 : Age = 21.0
Student Details --- Student Name = Udit : Score = 77.45 : Age = 21.0
Student Details --- Student Name = Shreya : Score = 80.76 : Age = 23.0
Student Details --- Student Name = Jerry : Score = 88.08 : Age = 24.0
Student Details --- Student Name = Zen : Score = 88.76 : Age = 24.0
Student Details --- Student Name = Akshita : Score = 98.09 : Age = 22.0
Student Details --- Student Name = Prabbhat : Score = 100.0 : Age = 23.0

Process finished with exit code 0
|
```

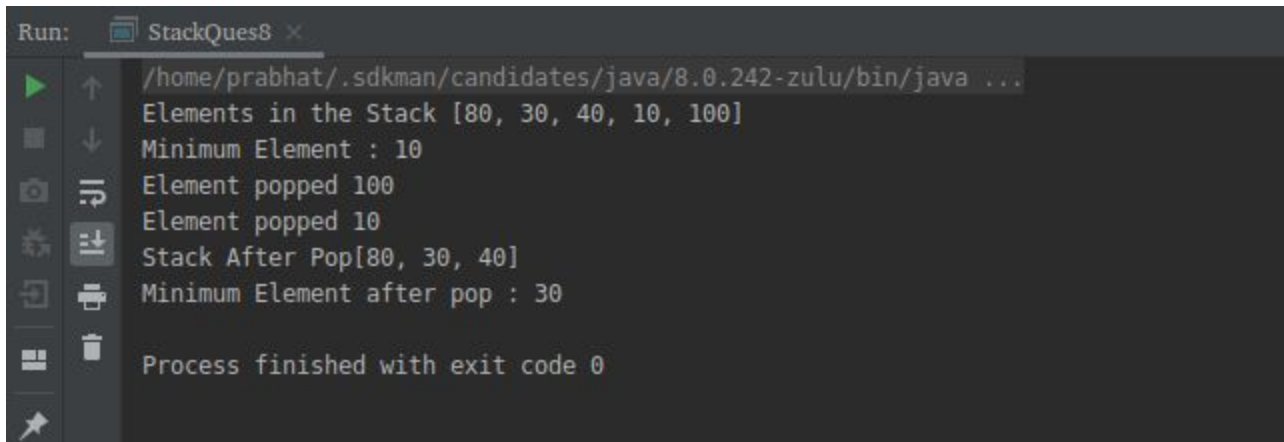
-
7. *Print the elements of an array in the decreasing frequency if 2 numbers have same frequency then print the one which came first.

OUTPUT(ArrayQues7.java)

```
Run: ArrayQues7 x
/home/prabhat/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Enter the Size of Array
10
2 4 5 2 3 7 5 4 3 2
2:3
4:2
5:2
3:2
7:1
7 4 4 5 5 3 3 2 2 2
Process finished with exit code 0
```

-
8. *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 $O(1)$)

OUTPUT(StackQues8.java)



```
Run: StackQues8 x
/home/prabhat/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Elements in the Stack [80, 30, 40, 10, 100]
Minimum Element : 10
Element popped 100
Element popped 10
Stack After Pop[80, 30, 40]
Minimum Element after pop : 30
Process finished with exit code 0
```